Robust and Simple ADMM Penalty Parameter Selection

McCann, Michael Thompson
Wohlberg, Brendt Egon

To be published in: IEEE Open Journal of Signal Processing

DOI to publisher's version: 10.1109/OJSP.2023.3349115

.

Robust and Simple ADMM Penalty Parameter Selection

Michael T. McCann, *Member, IEEE* and Brendt Wohlberg, *Fellow, IEEE*

[1]Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545

Corresponding author: Michael T. McCann (email: mccann@lanl.gov).

This research was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project numbers 20200061DR, 20230742CR, and 20230771DI.

ABSTRACT We present a new method for online selection of the penalty parameter for the alternating direction method of multipliers (ADMM) algorithm. ADMM is a widely used method for solving a range of optimization problems, including those that arise in signal and image processing. In its standard form, ADMM includes a scalar hyperparameter, known as the penalty parameter, which usually has to be tuned to achieve satisfactory empirical convergence. In this work, we develop a framework for analyzing the ADMM algorithm applied to a quadratic problem as an affine fixed point iteration. Using this framework, we develop a new method for automatically tuning the penalty parameter by detecting when it has become too large or small. We analyze this and several other methods with respect to their theoretical properties, i.e., robustness to problem transformations, and empirical performance on several optimization problems. Our proposed algorithm is based on a theoretical framework with clear, explicit assumptions and approximations, is theoretically covariant/invariant to problem transformations, is simple to implement, and exhibits competitive empirical performance.

INDEX TERMS convex optimization, ADMM, adaptive ADMM, penalty parameter, parameter selection

I. Introduction

Proximal algorithms are widely used for solving a variety of optimization problems in signal and image processing [1]. Of these, the alternating direction method of multipliers (ADMM) [2], [3] is particularly widely used due to its flexibility in addressing a wide range of problems. The ADMM algorithm solves optimization problems of the form

$$\mathop{\arg\min}_{\boldsymbol{x},\boldsymbol{z}} f(\boldsymbol{x}) + g(\boldsymbol{z}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{c}\,, \tag{1}$$

with variables[1] $\boldsymbol{x} \in \mathbb{R}^M$, $\boldsymbol{z} \in \mathbb{R}^N$; vector $\boldsymbol{c} \in \mathbb{R}^P$; matrices $\boldsymbol{A} \in \mathbb{R}^{P\times M}$ and $\boldsymbol{B} \in \mathbb{R}^{P\times N}$; convex functionals $f : \mathbb{R}^M \to \mathbb{R}$ and $g : \mathbb{R}^N \to \mathbb{R}$; and where $\arg\min f$ denotes any minimizer of f when the minimizer is not unique. (The notation used here is based on that of [5].) The ADMM iterates are

$$\boldsymbol{x}^{(k+1)} = \mathop{\arg\min}_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{\rho}{2}\left\| \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z}^{(k)} - \boldsymbol{c} + \frac{\boldsymbol{y}^{(k)}}{\rho}\right\|^2 \tag{2}$$

$$\boldsymbol{z}^{(k+1)} = \mathop{\arg\min}_{\boldsymbol{z}} g(\boldsymbol{z}) + \frac{\rho}{2}\left\| \boldsymbol{A}\boldsymbol{x}^{(k+1)} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{c} + \frac{\boldsymbol{y}^{(k)}}{\rho}\right\|^2 \tag{3}$$

$$\boldsymbol{y}^{(k+1)} = \boldsymbol{y}^{(k)} + \rho\big(\boldsymbol{A}\boldsymbol{x}^{(k+1)} + \boldsymbol{B}\boldsymbol{z}^{(k+1)} - \boldsymbol{c}\big)\,, \tag{4}$$

where $\rho \in \mathbb{R}$ is a positive scalar known as the *penalty parameter*, $\|\cdot\|$ denotes the ℓ_2 norm, and $\boldsymbol{y} \in \mathbb{R}^P$, known as the *dual variable*, [2] plays the role of the Lagrange multiplier for the constraint in (1).The iteration (2)-(4) can be shown to converge to a solution of (1) under a variety of conditions (see, e.g., [5, §3.2], [6], [7]).

In practice, the rate of convergence of ADMM algorithms is strongly dependent on the penalty parameter. Unfortunately, other than for a very specific set of problems [8], [9], [10, Sec. 5], there are no analytic results providing the optimal parameter choice. While a brute-force search for the best parameter—running the ADMM algorithm may times

[1] We only consider real-valued variable here. Problems with complex-valued variables may be expressed in this form by representing each variable as a real-valued vector containing its real and imaginary parts, but direct extension to complex-valued variables is also worthy of exploration [4].

[2] Note that, while it is common to introduce a *scaled dual variable* $\boldsymbol{u} = \rho^{-1}\boldsymbol{y}$ (e.g., [5, §3.1.1]), we retain the unscaled form since it makes the ρ dependency explicit and avoids the need for a rescaling of the dual variable when ρ is modified during the iterations of the ADMM algorithm. Appendix A describes how to use the proposed algorithm on the scaled form.

with different values of the parameter and keeping the best result—is straightforward to implement, it is computationally expensive and impractical for large-scale or real-time problems. Another solution is to adjust the penalty parameter as the algorithm is executed, sometimes called *online penalty parameter selection* or *adaptive ADMM*. The ADMM iterates are the same as those in (2)-(4), but all occurrences of ρ are replaced with an iteration-dependent value $\rho^{(k)}$, and a ρ update step

$$\rho^{(k+1)} = \phi\Big(\big(\rho^{(j)}, \boldsymbol{x}^{(j+1)}, \boldsymbol{z}^{(j+1)}, \boldsymbol{y}^{(j+1)}\big)_{j=0}^{k}\Big) , \tag{5}$$

is appended after the dual variable update. This update is defined in terms of a function $\phi : \mathbb{R} \times \mathbb{R}^M \times \mathbb{R}^N \times \mathbb{R}^P \times \cdots \to \mathbb{R}$ that selects a new penalty parameter based on all current and past penalty parameters, all current and past variables, and, implicitly, the problem definition f, g, $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{c}$.

A. Related Work

Several previous works address the problem of selecting the ADMM penalty parameter. A computationally efficient method has been proposed for brute-force evaluation over a large set of penalty parameter values [11], but its efficacy has only been explored for a very limited class of problems. Other works [8], [9], [10] have considered penalty parameter selection for ADMM applied specifically to quadratic programs, but these works involve explicit eigenvalue calculations that do not efficiently scale to large problems. A heuristic method for adapting the penalty parameter online [12] is widely used, but is sensitive to problem scaling, and can perform very poorly [13], as discussed in Section V. More recently, the interpretation of ADMM as Douglas-Rachford splitting (DRS) applied to the dual of (1) (see e.g. [14]) has been used to translate new step-size selection methods for DRS into penalty parameter selection methods for ADMM. The approach of [15], [16], is based on Barzilai-Borwein spectral step-size selection for DRS, while that of [17] is based on minimization of an upper bound on the spectral radius of the DRS iteration matrix. Both of these methods are discussed further in Section IV. Finally, [18] uses the dynamical systems approach developed in [19] to bound the convergence rate of ADMM, and proposes to select a fixed penalty parameter in advance to minimize this bound. This method involves technical assumptions that restrict which problems it can be used on, is complex to implement (because computing the convergence bound for each fixed penalty parameter involves searching for a scalar parameter that makes a certain parametric matrix inequality feasible), and, at least in the experiments in [18], did not appear to provide accurate penalty parameter selection for standard (unrelaxed) ADMM, which we study here.

Our work is particularly inspired by the analysis of quadratic problems in [8], the concept of rearranging ADMM into DRS from [15], the discussion of affine fixed points in [19], the analysis of linear DRS updates in [17], and the good empirical performance we observed when we implemented the method of [17].

B. Contributions and Outline

In this work, we propose a new mathematical framework and associated method for ADMM penalty parameter selection. This framework is distinct from previous work in that it focuses on determining whether the parameter is much too large or small rather than attempting to directly optimize it. In addition to motivating our method, our framework provides a new, unified perspective on previous penalty parameter selection methods [12], [15], [17]. Our method has not previously appeared in the literature, is simple to implement, and has theoretical advantages over each of the earlier methods. Computational experiments demonstrate that the proposed method provides competitive performance in practice across a variety of different problems.

The outline of the paper is as follows. The proposed framework is developed in Section II, followed by derivation of the proposed penalty parameter selection method in Section III. This framework is used to reinterpret the methods of [12], [15], and [17] in Section IV. The final theoretical component of the work is presented in Section V, with a discussion of how each method responds to transformations applied to the optimization problem, which is key to understanding the limitations of some of the existing methods. Experimental comparisons on several problems in are provided in Section VI, and conclusions are drawn in Section VII.

II. Penalty Parameter Selection Framework

In this section, we present the new framework for ADMM penalty parameter selection. The fundamental idea is to approximate the ADMM iterations locally (i.e., in the region of the current $\boldsymbol{x}^{(k)}$, $\boldsymbol{z}^{(k)}$, and $\boldsymbol{y}^{(k)}$) as an affine fixed point iteration $\boldsymbol{y}^{(k+1)} = \boldsymbol{H}_\rho \boldsymbol{y}^{(k)} + \boldsymbol{h}$ for $\boldsymbol{H}_\rho \in \mathbb{R}^{P \times P}$ and $\boldsymbol{h} \in \mathbb{R}^P$, where $\boldsymbol{H}_\rho$ depends on the penalty parameter ρ. The theory of affine fixed point iteration then allows us to view penalty parameter selection as selecting ρ to minimize the spectral radius of $\boldsymbol{H}_\rho$.

A. Iteration on $\boldsymbol{y}$

We show that the ADMM iterations (2)-(4) can be expressed as an iteration on $\boldsymbol{y}$ alone by recovering $\boldsymbol{x}$ and $\boldsymbol{z}$ from $\boldsymbol{y}$ at each iteration. This result is known in the literature, e.g., [14], [15], [17], but it is usually expressed as applying DRS to a dual version of problem (1), while our derivation avoids this complexity. We begin by using results from convex optimization to rewrite the optimization problem on $\boldsymbol{z}$ (3) as a function of $\boldsymbol{y}$.

According to the the first order optimality condition, the gradient of a smooth function is $\mathbf{0}$ at its local minima [20, Theorem 1.2.1]. For nonsmooth problems, a similar condition holds for the subgradient [20, Theorem 3.1.15]. The subgradient with respect to $\boldsymbol{z}$ of the functional in (3) is

$$\begin{aligned}
&\partial_{\boldsymbol{z}}\left(g(\boldsymbol{z})+\frac{\rho}{2}\left\|\boldsymbol{A}\boldsymbol{x}^{(k+1)}+\boldsymbol{B}\boldsymbol{z}-\boldsymbol{c}+\frac{\boldsymbol{y}^{(k)}}{\rho}\right\|^2\right)\\
&=\partial_{\boldsymbol{z}}g(\boldsymbol{z})+\nabla_{\boldsymbol{z}}\frac{\rho}{2}\left\|\boldsymbol{A}\boldsymbol{x}^{(k+1)}+\boldsymbol{B}\boldsymbol{z}-\boldsymbol{c}+\frac{\boldsymbol{y}^{(k)}}{\rho}\right\|^2\\
&=\partial_{\boldsymbol{z}}g(\boldsymbol{z})+\nabla_{\boldsymbol{z}}\frac{\rho}{2}\left\|\boldsymbol{B}\boldsymbol{z}-\left(-\boldsymbol{A}\boldsymbol{x}^{(k+1)}+\boldsymbol{c}-\frac{\boldsymbol{y}^{(k)}}{\rho}\right)\right\|^2\\
&=\partial_{\boldsymbol{z}}g(\boldsymbol{z})+\rho\left(\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{z}-\boldsymbol{B}^T\left(-\boldsymbol{A}\boldsymbol{x}^{(k+1)}+\boldsymbol{c}-\frac{\boldsymbol{y}^{(k)}}{\rho}\right)\right)\\
&=\partial_{\boldsymbol{z}}g(\boldsymbol{z})+\boldsymbol{B}^T\big(\boldsymbol{y}^{(k)}+\rho\big(\boldsymbol{A}\boldsymbol{x}^{(k+1)}+\boldsymbol{B}\boldsymbol{z}-\boldsymbol{c}\big)\big)\ . \qquad (6)
\end{aligned}$$

The optimality condition for (3) can therefore be written as [5, §3.3]

$$\begin{aligned}
\boldsymbol{0}&\in\partial_{\boldsymbol{z}}g(\boldsymbol{z}^{(k+1)})+\boldsymbol{B}^T\big(\underbrace{\boldsymbol{y}^{(k)}+\rho\big(\boldsymbol{A}\boldsymbol{x}^{(k+1)}+\boldsymbol{B}\boldsymbol{z}^{(k+1)}-\boldsymbol{c}\big)}_{\boldsymbol{y}^{(k+1)}}\big)\\
&\in\partial_{\boldsymbol{z}}g(\boldsymbol{z}^{(k+1)})+\boldsymbol{B}^T\boldsymbol{y}^{(k+1)}\ . \qquad (7)
\end{aligned}$$

We interpret (7) as the optimality condition for a new optimization problem on $\boldsymbol{z}$ (and for clarity shift the index from $k{+}1$ to k), which allows us to express $\boldsymbol{z}^{(k)}$ as a function of $\boldsymbol{y}^{(k)}$,

$$\boldsymbol{z}^{(k)}=G\big(\boldsymbol{y}^{(k)}\big)\quad G(\boldsymbol{w})=\operatorname*{arg\,min}_{\boldsymbol{z}}g(\boldsymbol{z})+\big(\boldsymbol{B}^T\boldsymbol{w}\big)^T\boldsymbol{z}\ , \qquad (8)$$

where $\boldsymbol{w}\in\mathbb{R}^P$. Note that (8) only holds for $k\geq 1$ because of the index shift.

Taking a similar approach for the $\boldsymbol{x}$ update (2), the subgradient with respect to $\boldsymbol{x}$ is

$$\begin{aligned}
&\partial_{\boldsymbol{x}}\left(f(\boldsymbol{x})+\frac{\rho}{2}\left\|\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{z}^{(k)}-\boldsymbol{c}+\frac{\boldsymbol{y}^{(k)}}{\rho}\right\|^2\right)\\
&=\partial_{\boldsymbol{x}}f(\boldsymbol{x})+\nabla_{\boldsymbol{x}}\frac{\rho}{2}\left\|\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{z}^{(k)}-\boldsymbol{c}+\frac{\boldsymbol{y}^{(k)}}{\rho}\right\|^2\\
&=\partial_{\boldsymbol{x}}f(\boldsymbol{x})+\nabla_{\boldsymbol{x}}\frac{\rho}{2}\left\|\boldsymbol{A}\boldsymbol{x}-\left(-\boldsymbol{B}\boldsymbol{z}^{(k)}+\boldsymbol{c}-\frac{\boldsymbol{y}^{(k)}}{\rho}\right)\right\|^2\\
&=\partial_{\boldsymbol{x}}f(\boldsymbol{x})+\rho\left(\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}-\boldsymbol{A}^T\left(-\boldsymbol{B}\boldsymbol{z}^{(k)}+\boldsymbol{c}-\frac{\boldsymbol{y}^{(k)}}{\rho}\right)\right)\\
&=\partial_{\boldsymbol{x}}f(\boldsymbol{x})+\boldsymbol{A}^T\big(\boldsymbol{y}^{(k)}+\rho\big(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{z}^{(k)}-\boldsymbol{c}\big)\big)\ . \qquad (9)
\end{aligned}$$

The optimality condition for (2) can therefore be written as

$$\begin{aligned}
\boldsymbol{0}&\in\partial_{\boldsymbol{x}}f\big(\boldsymbol{x}^{(k+1)}\big)+\boldsymbol{A}^T\big(\underbrace{\boldsymbol{y}^k+\rho(\boldsymbol{A}\boldsymbol{x}^{(k+1)}+\boldsymbol{B}\boldsymbol{z}^{(k)}-\boldsymbol{c})}_{\tilde{\boldsymbol{y}}^{(k+1)}}\big)\\
&\in\partial_{\boldsymbol{x}}f\big(\boldsymbol{x}^{(k+1)}\big)+\boldsymbol{A}^T\tilde{\boldsymbol{y}}^{(k+1)}\ , \qquad (10)
\end{aligned}$$

where we have introduced the notation $\tilde{\boldsymbol{y}}$ to denote the indicated not-quite-$\boldsymbol{y}$ quantity, which involves $\boldsymbol{z}^{(k)}$ instead of $\boldsymbol{z}^{(k+1)}$. Note that (10) is distinct from the standard dual optimality condition for $\boldsymbol{x}$ (see (3.9) in [5]) because it holds for all $\boldsymbol{x}^{(k)}$ rather than just the solution $\boldsymbol{x}^*$ and because it involves $\tilde{\boldsymbol{y}}$ rather than $\boldsymbol{y}$. Interpreting (10) as the optimality condition for a new optimization problem, we can obtain $\boldsymbol{x}^{(k)}$ from $\tilde{\boldsymbol{y}}^{(k)}$ by using an operation that we denote F,

$$\boldsymbol{x}^{(k)}=F\big(\tilde{\boldsymbol{y}}^{(k)}\big)\quad F(\boldsymbol{w})=\operatorname*{arg\,min}_{\boldsymbol{x}}f(\boldsymbol{x})+\big(\boldsymbol{A}^T\boldsymbol{w}\big)^T\boldsymbol{x}\ , \qquad (11)$$

where $\boldsymbol{w}\in\mathbb{R}^P$.

With these definitions in place, we are prepared to rewrite ADMM as an iteration on $\boldsymbol{y}^{(k)}$ alone. By definition, we have

$$\begin{aligned}
\boldsymbol{y}^{(k+1)}&=\boldsymbol{y}^{(k)}+\rho\big(\boldsymbol{A}\boldsymbol{x}^{(k+1)}+\boldsymbol{B}\boldsymbol{z}^{(k+1)}-\boldsymbol{c}\big) && (12)\\
\tilde{\boldsymbol{y}}^{(k+1)}&=\boldsymbol{y}^{(k)}+\rho\big(\boldsymbol{A}\boldsymbol{x}^{(k+1)}+\boldsymbol{B}\boldsymbol{z}^{(k)}-\boldsymbol{c}\big)\ . && (13)
\end{aligned}$$

Making use of F (11) and G (8), we have

$$\begin{aligned}
\boldsymbol{y}^{(k)}+\rho\boldsymbol{B}\boldsymbol{z}^{(k)}-\rho\boldsymbol{c}&=\tilde{\boldsymbol{y}}^{(k+1)}-\rho\boldsymbol{A}\boldsymbol{x}^{(k+1)}\\
&=\tilde{\boldsymbol{y}}^{(k+1)}-\rho\boldsymbol{A}F(\tilde{\boldsymbol{y}}^{(k+1)})\\
&=(\boldsymbol{I}-\rho\boldsymbol{A}F)\big(\tilde{\boldsymbol{y}}^{(k+1)}\big)\ , \qquad (14)
\end{aligned}$$

and therefore

$$\boldsymbol{y}^{(k)}+\rho\boldsymbol{B}G(\boldsymbol{y}^{(k)})-\rho\boldsymbol{c}=(\boldsymbol{I}-\rho\boldsymbol{A}F)\big(\tilde{\boldsymbol{y}}^{(k+1)}\big)\ . \qquad (15)$$

Similarly,

$$\begin{aligned}
\tilde{\boldsymbol{y}}^{(k+1)}-\rho\boldsymbol{B}\boldsymbol{z}^{(k)}&=\boldsymbol{y}^{(k+1)}-\rho\boldsymbol{B}\boldsymbol{z}^{(k+1)}\\
&=\boldsymbol{y}^{(k+1)}-\rho\boldsymbol{B}G\big(\boldsymbol{y}^{(k+1)}\big)\\
&=(\boldsymbol{I}-\rho\boldsymbol{B}G)\big(\boldsymbol{y}^{(k+1)}\big)\ , \qquad (16)
\end{aligned}$$

and therefore

$$\tilde{\boldsymbol{y}}^{(k+1)}-\rho\boldsymbol{B}G\big(\boldsymbol{y}^{(k)}\big)=(\boldsymbol{I}-\rho\boldsymbol{B}G)\big(\boldsymbol{y}^{(k+1)}\big)\ . \qquad (17)$$

Finally, we express $\boldsymbol{y}^{(k+1)}$ as a function of $\boldsymbol{y}^{(k)}$ by solving (15) for $\tilde{\boldsymbol{y}}^{(k+1)}$ and (17) for $\boldsymbol{y}^{(k+1)}$,

$$\begin{aligned}
\tilde{\boldsymbol{y}}^{(k+1)}&=(\boldsymbol{I}-\rho\boldsymbol{A}F)^{-1}\Big((\boldsymbol{I}+\rho\boldsymbol{B}G)\big(\boldsymbol{y}^{(k)}\big)-\rho\boldsymbol{c}\Big) \qquad (18)\\
\boldsymbol{y}^{(k+1)}&=(\boldsymbol{I}-\rho\boldsymbol{B}G)^{-1}\Big(\tilde{\boldsymbol{y}}^{(k+1)}-\rho\boldsymbol{B}G\big(\boldsymbol{y}^{(k)}\big)\Big)\ ,
\end{aligned}$$

where $M^{-1}(\boldsymbol{x})$ for generic function M and vector $\boldsymbol{x}$ denotes a vector $\boldsymbol{y}$ such that $M(\boldsymbol{y})=\boldsymbol{x}$, which may not be unique. These inverses exist whenever the ADMM iterations (2)-(4) are well defined because, by the preceding arguments, ADMM generates sequences $\{\boldsymbol{x}^{(k)}\}$, $\{\boldsymbol{z}^{(k)}\}$, and $\{\boldsymbol{y}^{(k)}\}$ (and therefore implicitly $\{\tilde{\boldsymbol{y}}^{(k)}\}$ via the definition in (13)) that satisfy (15) and (17), which are the equations that the inverses in (18) solve.

Note that while there appear to be two state variables, $\tilde{\boldsymbol{y}}^{(k)}$ is a merely a notational convenience. The iteration can be written without $\tilde{\boldsymbol{y}}^{(k)}$ by substituting the first line of (18) into the second. Finally, it is worth noting that our derivation of (18) represents a novel approach to demonstrating the mathematical equivalence of ADMM and DRS.

B. *Affine Fixed Point Iteration*

We now analyze a quadratic problem, allowing us to express ADMM as an affine fixed point iteration, which is a critical component in the derivation of our penalty parameter selection rule. While ADMM is not typically used to solve such quadratic problems, they will act as local approximations of the actual problems of interest.

Consider problem (1) with quadratic f and g,

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{q}^T\boldsymbol{x} \qquad g(\boldsymbol{z}) = \frac{1}{2}\boldsymbol{z}^T\boldsymbol{R}\boldsymbol{z} + \boldsymbol{r}^T\boldsymbol{z}\ , \quad (19)$$

where $\boldsymbol{Q} \in \mathbb{R}^{M\times M}$, $\boldsymbol{R} \in \mathbb{R}^{N\times N}$, $\boldsymbol{q} \in \mathbb{R}^M$, and $\boldsymbol{r} \in \mathbb{R}^N$. To make f and g convex and ensure that the problem has a solution, we assume $\boldsymbol{Q}$ and $\boldsymbol{R}$ are symmetric positive definite matrices. This assumption further implies that they cannot have a nontrivial null space (because, e.g., $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{0}$ implies $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{0}$) and are therefore invertible.

Then, by definitions (11) and (8),

$$\begin{aligned} F(\boldsymbol{w}) &= -\boldsymbol{Q}^{-1}(\boldsymbol{A}^T\boldsymbol{w} + \boldsymbol{q}) \\ G(\boldsymbol{w}) &= -\boldsymbol{R}^{-1}(\boldsymbol{B}^T\boldsymbol{w} + \boldsymbol{r})\ , \end{aligned} \quad (20)$$

Since F and G always appear in (18) as terms $\boldsymbol{A}F$ and $\boldsymbol{B}G$ respectively, we introduce the notation

$$\mathbb{F} = \boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^T \qquad \mathbb{G} = \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T \quad (21)$$

for the linear parts of $\boldsymbol{A}F$ and $\boldsymbol{B}G$ respectively, where we use the blackboard bold notation to clearly differentiate the matrices $\mathbb{F}$ and $\mathbb{G}$ from the closely-related functions F and G.

We can substitute the first line of (18) into the second and rearrange to express ADMM as

$$\begin{aligned} \boldsymbol{y}^{(k+1)} &= (\boldsymbol{I}+\rho\mathbb{G})^{-1}\left((\boldsymbol{I}+\rho\mathbb{F})^{-1}(\boldsymbol{I}-\rho\mathbb{G})\boldsymbol{y}^{(k)} + \rho\mathbb{G}\boldsymbol{y}^{(k)}\right) + \boldsymbol{h} \\ &= (\boldsymbol{I}+\rho\mathbb{G})^{-1}\left(((\boldsymbol{I}+\rho\mathbb{F})^{-1}(\boldsymbol{I}-\rho\mathbb{G}) + \rho\mathbb{G})\boldsymbol{y}^{(k)}\right) + \boldsymbol{h} \\ &= (\boldsymbol{I}+\rho\mathbb{G})^{-1}(\boldsymbol{I}+\rho\mathbb{F})^{-1}\left(((\boldsymbol{I}-\rho\mathbb{G}) + (\boldsymbol{I}+\rho\mathbb{F})\rho\mathbb{G})\boldsymbol{y}^{(k)}\right) \\ &\quad + \boldsymbol{h} \\ &= \underbrace{(\boldsymbol{I}+\rho\mathbb{G})^{-1}(\boldsymbol{I}+\rho\mathbb{F})^{-1}(\boldsymbol{I}+\rho^2\mathbb{F}\mathbb{G})}_{\boldsymbol{H}_\rho}\boldsymbol{y}^{(k)} + \boldsymbol{h}\ , \end{aligned} \quad (22)$$

where $\boldsymbol{H}_\rho$ is a matrix that depends on ρ and $\boldsymbol{h}$ is a constant vector that is unimportant for what follows. Note that $\tilde{\boldsymbol{y}}$ does not appear because it is only a notational convenience and can be expressed in terms of $\boldsymbol{y}$.

We now have ADMM expressed as the affine fixed point iteration

$$\boldsymbol{y}^{(k+1)} = \boldsymbol{H}_\rho\boldsymbol{y}^{(k)} + \boldsymbol{h}\ . \quad (23)$$

We define the fixed point, $\boldsymbol{y}^*$, when it exists, by

$$\boldsymbol{y}^* = \boldsymbol{H}_\rho\boldsymbol{y}^* + \boldsymbol{h}\ . \quad (24)$$

It is worth emphasizing that, while $\boldsymbol{H}_\rho$ and $\boldsymbol{h}$ depend on ρ, $\boldsymbol{y}^*$ does not. The fixed point $\boldsymbol{y}^*$ is independent of ρ because the iteration (23) is equivalent to ADMM applied to a quadratic problem with a unique solution. Because ADMM converges to a solution of this problem [6] and the solution is unique, it cannot depend on ρ.

We can now relate the convergence rate of ADMM to the spectral radius (the magnitude of the eigenvalue with the largest magnitude) Define the error at iterate k by

$$\boldsymbol{\epsilon}^{(k)} = \boldsymbol{y}^{(k)} - \boldsymbol{y}^*\ . \quad (25)$$

From this definition, (23), and (24) we have

$$\boldsymbol{\epsilon}^{(k)} = \boldsymbol{H}_\rho(\boldsymbol{y}^{(k-1)} + \boldsymbol{h}) - (\boldsymbol{H}_\rho\boldsymbol{y}^* + \boldsymbol{h}) = \boldsymbol{H}_\rho\boldsymbol{\epsilon}^{(k-1)} \quad (26)$$

and so by induction,

$$\boldsymbol{\epsilon}^{(k)} = {\boldsymbol{H}_\rho}^k\boldsymbol{\epsilon}^{(0)}\ . \quad (27)$$

Denote the spectral radius of $\boldsymbol{H}_\rho$ by $r(\boldsymbol{H}_\rho)$. Gelfand's formula [21, Theorem 8] states

$$r(\boldsymbol{H}_\rho) \le \|{\boldsymbol{H}_\rho}^k\|^{\frac{1}{k}} \quad \text{and} \quad r(\boldsymbol{H}_\rho) = \lim_{k\to\infty}\|{\boldsymbol{H}_\rho}^k\|^{\frac{1}{k}}\ , \quad (28)$$

which, as noted in [19, §2.2], implies that for any $e > 0$, there is a k large enough that

$$\|\boldsymbol{\epsilon}^{(k)}\| \le (r(\boldsymbol{H}_\rho) + e)^k\|\boldsymbol{\epsilon}^{(0)}\|\ . \quad (29)$$

The convergence rate of the fixed point is therefore determined by the spectral radius of $\boldsymbol{H}_\rho$. Note that this bound implies that $r(\boldsymbol{H}_\rho) < 1$ is sufficient for convergence of the fixed point.

We now derive a further consequence of the affine fixed point which we use in the next section. The expression for the error (27) corresponds to a power iteration of $\boldsymbol{H}_\rho$. So, assuming that the maximal eigenvalue of $\boldsymbol{H}_\rho$ is real,[3] as k grows, $\boldsymbol{\epsilon}^{(k)}$ converges to a maximal eigenvector of $\boldsymbol{H}_\rho$ [22, §7.3.1], i.e.,

$$\lim_{k\to\infty}\frac{{\boldsymbol{\epsilon}^{(k)}}^T\boldsymbol{H}_\rho\boldsymbol{\epsilon}^{(k)}}{{\boldsymbol{\epsilon}^{(k)}}^T\boldsymbol{\epsilon}^{(k)}} = r(\boldsymbol{H}_\rho) \quad (30)$$

and

$$\lim_{k\to\infty}\boldsymbol{H}_\rho\boldsymbol{\epsilon}^{(k)} = r(\boldsymbol{H}_\rho)\boldsymbol{\epsilon}^{(k)}\ . \quad (31)$$

Therefore, for k sufficiently large and $\Delta k > 0$,

$$\begin{aligned} \boldsymbol{y}^{(k+\Delta k)} - \boldsymbol{y}^{(k)} &= \left(\boldsymbol{y}^* + \boldsymbol{\epsilon}^{(k+\Delta k)}\right) - \left(\boldsymbol{y}^* + \boldsymbol{\epsilon}^{(k)}\right) \\ &= \boldsymbol{\epsilon}^{(k+\Delta k)} - \boldsymbol{\epsilon}^{(k)} \\ &= {\boldsymbol{H}_\rho}^{\Delta k}\boldsymbol{\epsilon}^{(k)} - \boldsymbol{\epsilon}^{(k)} \\ &\approx \left(r(\boldsymbol{H}_\rho)^{\Delta k} - 1\right)\boldsymbol{\epsilon}^{(k)}\ , \end{aligned} \quad (32)$$

where the last line follows from $\boldsymbol{\epsilon}^{(k)}$ approximately being a maximal eigenvector of $\boldsymbol{H}_\rho$. This equation implies that $\boldsymbol{y}^{(k+\Delta k)} - \boldsymbol{y}^{(k)}$ is colinear with $\boldsymbol{\epsilon}^{(k)}$ and is therefore also approximately a maximal eigenvector of $\boldsymbol{H}_\rho$.

Let $\boldsymbol{v}_\rho$ denote a maximal eigenvector of $\boldsymbol{H}_\rho$ with corresponding eigenvalue λ_ρ. If $|\lambda_\rho| > 1$ then (29) implies that the error would grow with each iteration, so $|\lambda_\rho|$ must be smaller than one for the fixed point iteration, and therefore ADMM, to converge. In addition, we expect $|\lambda_\rho|$ to be close to one because convergence of ADMM typically takes at least tens of iterations, implying that $|\lambda_\rho| = r(\boldsymbol{H}_\rho)$ in (29) is usually not much smaller than unity.

III. Proposed Penalty Parameter Selection Method

With these results in place, we are prepared to derive our penalty parameter selection method. Our approach is motivated by the empirical result (see e.g. [23], [18], [15]) that there is typically a single optimal penalty parameter for each problem, with convergence degrading as ρ moves away

[3] In general, $\boldsymbol{H}_\rho$ may have complex eigenvalues and eigenvectors. In Section III-B, we argue that when ρ is far from its optimal value, $\boldsymbol{H}_\rho$ is approximated by a matrix with real eigenvalues.

from this value. Here, we derive approximations that explain this monotone behavior and use them to propose a rule for selecting a ρ that is not too far from the optimal value.

While the previous section represents a synthesis of results from multiple sources, what follows is, to the best of our knowledge, novel.

A. Dependence of Spectral Radius on Penalty Parameter

In the previous section, we determined that, for a quadratic problem, the optimal ρ is the one that minimizes the spectral radius of the iteration matrix $\boldsymbol{H}_\rho$. Therefore, we would like to know how the eigenvalues of $\boldsymbol{H}_\rho$ depend on ρ. For small quadratic problems, we can form $\boldsymbol{H}_\rho$ explicitly and compute these eigenvalues directly for a range of ρ values, but this is impractical for larger problems. More importantly, though, since our primary interest is in more general problems, we would like to be able to determine this dependence using variables and operators that are not specific to the quadratic problem so that we can avoid having to explicitly fit quadratic approximations. Unfortunately, the eigenvalues of $\boldsymbol{H}_\rho$ vary with ρ, and the spectral radius can change in a complex way, as illustrated in Fig. 1. The same figure suggests that the spectral radius depends much more simply on ρ as ρ moves away from its optimal value: when ρ is too large, $r(\boldsymbol{H}_\rho)$ increases monotonically with ρ; when ρ is too small, $r(\boldsymbol{H}_\rho)$ decreases monotonically with ρ.

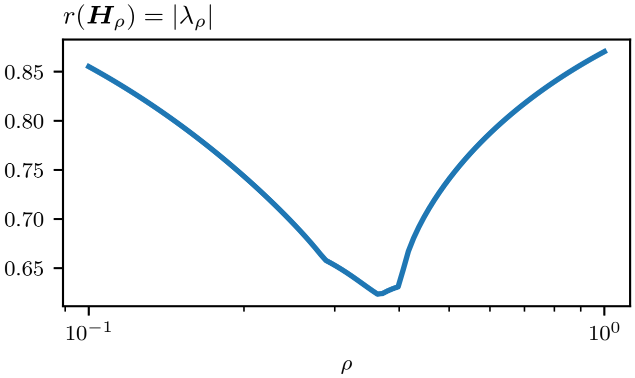


FIGURE 1: Spectral radius of the affine iteration matrix for a sum of quadratics problem (see Section VI-B for details). The spectral radius is a complex function of ρ, however its behavior becomes monotone as ρ grows small or large relative to the location of the minimum.

Recalling the definition of the affine iteration matrix $\boldsymbol{H}_\rho$ (22), we have that for maximal[4] eigenvector $\boldsymbol{v}_\rho$ with eigenvalue λ_ρ (assumed to be real, as previously noted),

$$(\boldsymbol{I}+\rho\mathbb{G})^{-1}(\boldsymbol{I}+\rho\mathbb{F})^{-1}(\boldsymbol{I}+\rho^2\mathbb{F}\mathbb{G})\boldsymbol{v}_\rho = \lambda_\rho \boldsymbol{v}_\rho \ , \quad (33)$$

and therefore by premultiplying by $\boldsymbol{v}_\rho^T(\boldsymbol{I}+\rho\mathbb{F})(\boldsymbol{I}+\rho\mathbb{G})$ and dividing,

$$\lambda_\rho = \frac{\boldsymbol{v}_\rho^T(\boldsymbol{I}+\rho^2\mathbb{F}\mathbb{G})\boldsymbol{v}_\rho}{\boldsymbol{v}_\rho^T\left(\boldsymbol{I}+\rho(\mathbb{G}+\mathbb{F})+\rho^2\mathbb{F}\mathbb{G}\right)\boldsymbol{v}_\rho} \ . \quad (34)$$

If we find the maximal eigenvector at a particular $\rho=\rho_0$ and assume that it is not changing too much with ρ, we have an approximation of the form

$$\lambda_{\rho,\rho_0} = \frac{\boldsymbol{v}_{\rho_0}^T(\boldsymbol{I}+\rho^2\mathbb{F}\mathbb{G})\boldsymbol{v}_{\rho_0}}{\boldsymbol{v}_{\rho_0}^T\left(\boldsymbol{I}+\rho(\mathbb{G}+\mathbb{F})+\rho^2\mathbb{F}\mathbb{G}\right)\boldsymbol{v}_{\rho_0}} \quad (35)$$

which is a rational polynomial in ρ. We plot two of these approximations in Fig. 2, which shows excellent agreement between the approximations and the true spectral radius in the area around ρ_0.

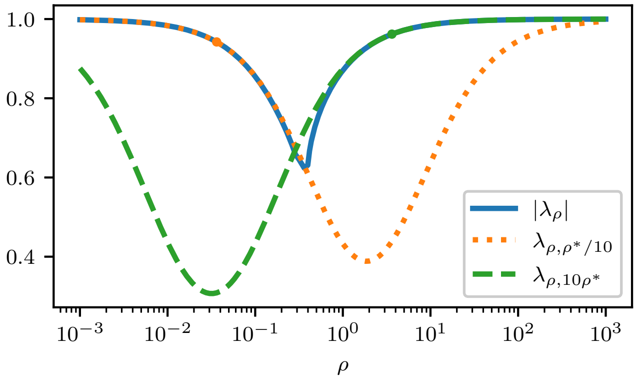


FIGURE 2: Spectral radius of the affine iteration matrix for a sum of quadratics problem (see Section VI-B for details) along with the approximation (35) for two different ρ_0 values. Each approximation shows excellent agreement with the spectral radius in the area around ρ_0, marked with a colored dot.

B. Proposed Penalty Parameter Selection Method

The core idea of our penalty parameter selection method, which we call the **spectral radius approximation (SRA) method**, is that, when ρ gets either much larger or much smaller than its optimal value, simple, monotone approximations for the relationship between λ_ρ and ρ hold. We select ρ so as to avoid the regimes in which these approximations hold. To make this practical, the determination of when the approximations hold should be made using quantities that can be cheaply computed from the working variables of ADMM, avoiding explicitly forming large matrices and computing their eigenvalues. Of the terms $\boldsymbol{v}_\rho$, $\mathbb{F}\boldsymbol{v}_\rho$, $\mathbb{G}\boldsymbol{v}_\rho$, and $\mathbb{F}\mathbb{G}\boldsymbol{v}_\rho$ that play a role in the previous section, we only know how to compute $\boldsymbol{v}_\rho$ and $\mathbb{G}\boldsymbol{v}_\rho$ in this way. We now derive efficient estimates for these terms.

As discussed before, $\boldsymbol{y}^{(k+\Delta k)} - \boldsymbol{y}^{(k)}$ is approximately a maximal eigenvalue of $\boldsymbol{H}_\rho$, so $\boldsymbol{v}_\rho \approx \boldsymbol{y}^{(k+\Delta k)} - \boldsymbol{y}^{(k)}$. If we set $\Delta k = 1$, we have

$$\boldsymbol{v}_\rho \approx \boldsymbol{y}^{(k+1)} - \boldsymbol{y}^{(k)} = \rho(\boldsymbol{A}\boldsymbol{x}^{(k+1)} + \boldsymbol{B}\boldsymbol{z}^{(k+1)} - \boldsymbol{c}) \ , \quad (36)$$

[4] The following equations actually hold for any eigenvalue/vector pair, but we are specifically concerned with their consequences for the maximal one.

which has the benefit of not requiring the storage of past $\boldsymbol{y}$ values. To compute $\mathbb{G}\boldsymbol{v}_\rho$, we can substitute the expression we just derived for $\boldsymbol{v}_\rho$ and use the definitions of $\mathbb{G}$ and G. From (20) we have that

$$G(\boldsymbol{u}) - G(\boldsymbol{w}) = -\boldsymbol{R}^{-1}\boldsymbol{B}^T(\boldsymbol{u} - \boldsymbol{w}) \tag{37}$$

for arbitrary vectors $\boldsymbol{u}$ and $\boldsymbol{w}$, and therefore

$$\begin{aligned}\mathbb{G}\boldsymbol{v}_\rho &\approx \mathbb{G}\big(\boldsymbol{y}^{(k+\Delta k)} - \boldsymbol{y}^{(k)}\big)\\ &= \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\big(\boldsymbol{y}^{(k+\Delta k)} - \boldsymbol{y}^{(k)}\big)\\ &= -\boldsymbol{B}\big(G\big(\boldsymbol{y}^{(k+\Delta k)}\big) - G\big(\boldsymbol{y}^{(k)}\big)\big)\\ &= -\boldsymbol{B}\big(\boldsymbol{z}^{(k+\Delta k)} - \boldsymbol{z}^{(k)}\big)\ . \end{aligned} \tag{38}$$

Again, it is convenient to use the most recent iterate,

$$\rho\mathbb{G}\boldsymbol{v}_\rho \approx -\rho\boldsymbol{B}\big(\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)}\big)\ . \tag{39}$$

Case 1: ρ_0 small. In the case that ρ_0 is small enough that $\|\rho_0\mathbb{G}\boldsymbol{v}_{\rho_0}\|/\|\boldsymbol{v}_{\rho_0}\| \ll 1$, then in the neighborhood of $\rho = \rho_0$, it makes sense to approximate $(\boldsymbol{I} + \rho\mathbb{G})\boldsymbol{v}_{\rho_0}$ with $\boldsymbol{v}_{\rho_0}$. We then have[5]

$$\begin{aligned}&\big(\boldsymbol{I} + \rho(\mathbb{G} + \mathbb{F}) + \rho^2\mathbb{F}\mathbb{G}\big)\boldsymbol{v}_{\rho_0} && (40)\\ &\qquad= \big((\boldsymbol{I} + \rho\mathbb{G}) + \rho\mathbb{F}(\boldsymbol{I} + \rho\mathbb{G})\big)\boldsymbol{v}_{\rho_0}\\ &\qquad\approx \big((\boldsymbol{I} + \rho\mathbb{G}) + \rho\mathbb{F}\big)\boldsymbol{v}_{\rho_0}\\ &\qquad= \big(\boldsymbol{I} + \rho(\mathbb{F} + \mathbb{G})\big)\boldsymbol{v}_{\rho_0} && (41)\end{aligned}$$

so that

$$\big(\boldsymbol{I} + \rho^2\mathbb{F}\mathbb{G}\big)\boldsymbol{v}_{\rho_0} \approx \boldsymbol{v}_{\rho_0}\ . \tag{42}$$

Applying this approximation to (35) results in

$$\lambda_{\rho,\rho_0} \approx \lambda^{\text{small}}_{\rho,\rho_0} = \frac{\boldsymbol{v}^T_{\rho_0}\boldsymbol{v}_{\rho_0}}{\boldsymbol{v}^T_{\rho_0}\big(\boldsymbol{I} + \rho(\mathbb{G} + \mathbb{F})\big)\boldsymbol{v}_{\rho_0}}\ , \tag{43}$$

which is a decreasing function of ρ.[6]

Note that applying the same approximation to (33) shows that $\lambda^{\text{small}}_{\rho,\rho_0}$ is an eigenvalue of the matrix

$$\boldsymbol{I} + \rho(\mathbb{G} + \mathbb{F})\ , \tag{44}$$

which is symmetric because $\mathbb{F}$ and $\mathbb{G}$ are symmetric. The eigenvalue $\lambda^{\text{small}}_{\rho,\rho_0}$ is therefore real, justifying the assumption in Section II-B that $\boldsymbol{H}_\rho$ has real eigenvalues.

Case 2: ρ_0 large. In the case that ρ_0 is large enough that $\|\rho_0\mathbb{G}\boldsymbol{v}_{\rho_0}\|/\|\boldsymbol{v}_{\rho_0}\| \gg 1$, then in the neighborhood of $\rho = \rho_0$, it makes sense to approximate $(\boldsymbol{I} + \rho\mathbb{G})\boldsymbol{v}_{\rho_0}$ with $\rho\mathbb{G}\boldsymbol{v}_{\rho_0}$. We then have

$$\begin{aligned}\big(\boldsymbol{I} + \rho(\mathbb{G}{+}\mathbb{F}) + \rho^2\mathbb{F}\mathbb{G}\big)\boldsymbol{v}_{\rho_0} &= \big((\boldsymbol{I}{+}\rho\mathbb{G}) + \rho\mathbb{F} + \rho^2\mathbb{F}\mathbb{G}\big)\boldsymbol{v}_{\rho_0}\\ &\approx \big(\rho\mathbb{G} + \rho\mathbb{F} + \rho^2\mathbb{F}\mathbb{G}\big)\boldsymbol{v}_{\rho_0}\\ &= \big(\rho(\mathbb{G}{+}\mathbb{F}) + \rho^2\mathbb{F}\mathbb{G}\big)\boldsymbol{v}_{\rho_0}\ ,\end{aligned} \tag{45}$$

and therefore

$$\big(\boldsymbol{I} + \rho^2\mathbb{F}\mathbb{G}\big)\boldsymbol{v}_{\rho_0} \approx \rho^2\mathbb{F}\mathbb{G}\boldsymbol{v}_{\rho_0}\ . \tag{46}$$

Applying this approximation to (35) results in

$$\lambda_{\rho,\rho_0} \approx \lambda^{\text{large}}_{\rho,\rho_0} = \frac{\boldsymbol{v}^T_{\rho_0}\rho\mathbb{F}\mathbb{G}\boldsymbol{v}_{\rho_0}}{\boldsymbol{v}^T_{\rho_0}\big((\mathbb{G} + \mathbb{F}) + \rho\mathbb{F}\mathbb{G}\big)\boldsymbol{v}_{\rho_0}}\ , \tag{47}$$

which is an increasing function of ρ.[7]

Note that applying the same approximation to (33) shows that $\lambda^{\text{large}}_{\rho,\rho_0}$ is an eigenvalue of the matrix

$$((\mathbb{G} + \mathbb{F}) + \rho\mathbb{F}\mathbb{G})^{-1}\rho\mathbb{F}\mathbb{G}\ . \tag{50}$$

We have numerical evidence that this matrix has real eigenvalues when $\mathbb{F}$ and $\mathbb{G}$ are positive semidefinite, justifying the assumption that $\boldsymbol{H}_\rho$ has real eigenvalues made in Section II-B, but we have not found a proof that this property holds.

Proposed method. The proposed method is based on avoiding either of the previously mentioned cases, i.e., we want to avoid either

$$\|\boldsymbol{v}_{\rho_0}\| \ll \|\rho\mathbb{G}\boldsymbol{v}_{\rho_0}\| \quad \text{or} \quad \|\boldsymbol{v}_{\rho_0}\| \gg \|\rho\mathbb{G}\boldsymbol{v}_{\rho_0}\|\ . \tag{51}$$

Using the expressions we just derived, this is equivalent to avoiding

$$\big\|\boldsymbol{y}^{(k+1)} - \boldsymbol{y}^{(k)}\big\| \ll \rho\big\|\boldsymbol{B}(\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)})\big\| \tag{52}$$

or

$$\rho\|\boldsymbol{B}(\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)})\| \ll \|\boldsymbol{y}^{(k+1)} - \boldsymbol{y}^{(k)}\|\ . \tag{53}$$

While there are several possible ways to avoid these cases, we propose to simply select ρ so that the left and right sides of these inequality are equal,

$$\big\|\boldsymbol{y}^{(k+1)} - \boldsymbol{y}^{(k)}\big\| = \rho\big\|\boldsymbol{B}(\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)})\big\|\ , \tag{54}$$

giving the rule

$$\rho^{(k+1)}_{\text{SRA}} = \frac{\big\|\boldsymbol{y}^{(k+1)} - \boldsymbol{y}^{(k)}\big\|}{\big\|\boldsymbol{B}(\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)})\big\|}\ . \tag{55}$$

An empirical validation of this rule for the quadratic problem described in Section VI-B is presented in Fig. 3.

It may be surprising that the proposed rule does not involve $\boldsymbol{x}$ or $\boldsymbol{A}$. We designed the rule in this way because, as we have just shown, there is a way to express $\mathbb{G}\boldsymbol{v}$ in terms of $\boldsymbol{z}$ but there is not (to our knowledge) a symmetrical way to efficiently express $\mathbb{F}\boldsymbol{v}$ in terms of $\boldsymbol{x}$, c.f. (8) and (11).

[5] While an additional $\rho\mathbb{G}\boldsymbol{v}_{\rho_0}$ term could be removed, we do not do so for symmetry with the next approximation.

[6] By differentiating with respect to ρ, we know that (43) is a decreasing function when $\boldsymbol{v}^T_{\rho_0}\boldsymbol{v}_{\rho_0} + \boldsymbol{v}^T_{\rho_0}\rho(\mathbb{G}{+}\mathbb{F})\boldsymbol{v}_{\rho_0}$ is positive. It is positive because $\boldsymbol{v}^T_{\rho_0}\boldsymbol{v}_{\rho_0} = \|\boldsymbol{v}_{\rho_0}\|^2 \geq 0$ and $\mathbb{F}$ and $\mathbb{G}$ are positive semidefinite: We have $\boldsymbol{u}^T\mathbb{F}\boldsymbol{u} = \boldsymbol{u}^T\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^T\boldsymbol{u} = (\boldsymbol{A}^T\boldsymbol{u})^T\boldsymbol{Q}^{-1}(\boldsymbol{A}^T\boldsymbol{u}) \geq 0$ because $\boldsymbol{Q}$ is symmetric positive definite by assumption, and therefore so is $\boldsymbol{Q}^{-1}$. A similar argument holds for $\mathbb{G}$.

[7] By differentiating with respect to ρ, we know that (47) is an increasing function when $\boldsymbol{v}^T_{\rho_0}\mathbb{F}\mathbb{G}\boldsymbol{v}_{\rho_0} + \boldsymbol{v}^T_{\rho_0}\rho(\mathbb{G} + \mathbb{F})\boldsymbol{v}_{\rho_0}$ is positive. The term $\boldsymbol{v}^T_{\rho_0}\rho(\mathbb{G}{+}\mathbb{F})\boldsymbol{v}_{\rho_0}$ is nonnegative because $\mathbb{F}$ and $\mathbb{G}$ are positive semidefinite. The term $\boldsymbol{v}^T_{\rho_0}\mathbb{F}\mathbb{G}\boldsymbol{v}_{\rho_0}$ is nonnegative because the eigenvector equation (33) implies that

$$((1 - \lambda_{\rho_0})\boldsymbol{I} - \rho\lambda_{\rho_0}(\mathbb{G} + \mathbb{F}) + \rho^2(1 - \lambda_{\rho_0})\mathbb{F}\mathbb{G})\boldsymbol{v}_{\rho_0} = 0\ , \tag{48}$$

and therefore in the ρ_0 large case we are currently considering,

$$(-\rho\lambda_{\rho_0}(\mathbb{G} + \mathbb{F}) + \rho^2(1 - \lambda_{\rho_0})\mathbb{F}\mathbb{G})\boldsymbol{v}_{\rho_0} \approx 0\ , \tag{49}$$

which implies $\mathbb{F}\mathbb{G}\boldsymbol{v}_{\rho_0}$ is a positive scalar multiple of $(\mathbb{G} + \mathbb{F})\boldsymbol{v}_{\rho_0}$, so $\boldsymbol{v}^T_{\rho_0}\rho(\mathbb{G} + \mathbb{F})\boldsymbol{v}_{\rho_0} \geq 0$ implies $\boldsymbol{v}^T_{\rho_0}\mathbb{F}\mathbb{G}\boldsymbol{v}_{\rho_0} \geq 0$.

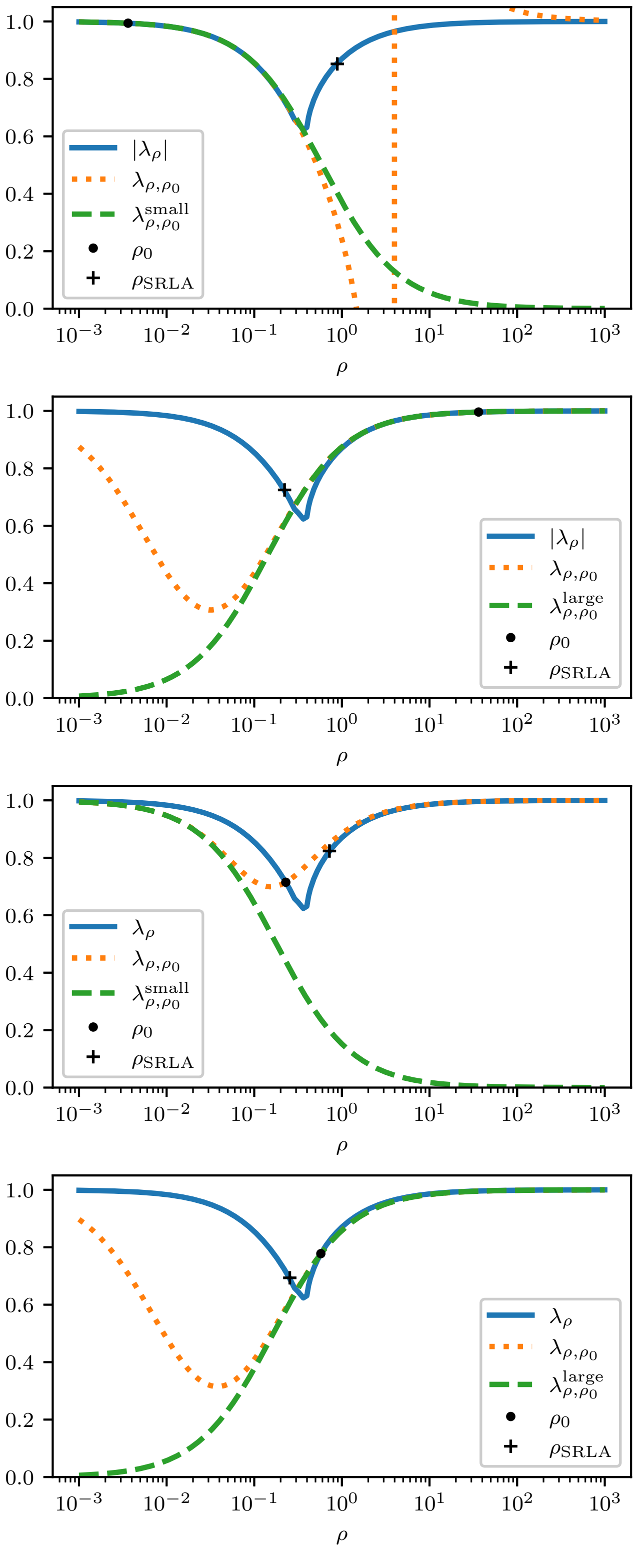


FIGURE 3: The spectral radius, the approximations (35), (43), and (47), and the proposed ρ for four different ρ_0's. When ρ_0 is far from ρ^* (top two plots), the approximations are accurate and ρ_{SRA} is close to ρ^*. When ρ_0 is close to ρ^* (bottom two plots), our justification for the approximations no longer holds, but ρ_{SRA} remains close to ρ^*.

There are several details to consider when implementing a complete penalty parameter selection method based on (55). We postpone these implementation details until Section VI-A and Algorithm 1.

C. Application to Nonquadratic Problems

Our derivation of the rule (55) was based on analyzing ADMM for a quadratic problem, but since it only involves $\boldsymbol{y}$, $\boldsymbol{B}$, and $\boldsymbol{z}$, it is possible, in a computational sense at least, to apply it to any ADMM algorithm. But is it reasonable to expect it to work well? The empirical results reported in Section VI indicate that it does, and previous ADMM penalty parameter selection methods [15], [17], which are also based on analysis of a quadratic problem, also report good empirical performance on general problems.

We believe that these algorithms generalize well because convex functions can be *locally* well-approximated by quadratics. Over a few iterations—and especially when ρ is not at its optimal value—the variables $\boldsymbol{x}$ and $\boldsymbol{z}$ do not change rapidly as a function of k, and therefore ADMM on f and g is similar to ADMM on a quadratic approximation of f and g in the local region of $\boldsymbol{x}^{(k)}$, $\boldsymbol{z}^{(k)}$. As $\boldsymbol{x}$ and $\boldsymbol{z}$ change over many iterations, the same argument can be made about the new neighborhood. For example, the total variation problem (121) is exactly quadratic in each region where the argument of the ℓ_1 norm does not change signs. Such iterative, local approximations are widely used in optimization, e.g., in Newton and quasi-Newton methods.

IV. New Interpretations of Existing Penalty Parameter Selection Methods

We now interpret several state-of-the-art ADMM penalty parameter selection methods from the literature in terms of the proposed framework. We emphasize that this unified perspective is distinct from the ones used to originally derive each method.

A. Residual Balancing Method

Residual balancing (RB) [12] is a straightforward and widely used (see, e.g. [24], [25], [26], [27], [28], [29]) approach to ADMM penalty parameter selection. It is based on an attempt to balance the norms of the primal and dual residuals at iteration k, which are defined as [5, §3.3]

$$\boldsymbol{r}^{(k)} = \boldsymbol{A}\boldsymbol{x}^{(k)} + \boldsymbol{B}\boldsymbol{z}^{(k)} - \boldsymbol{c} \tag{56}$$

and

$$\boldsymbol{s}^{(k)} = \rho \boldsymbol{A}^T \boldsymbol{B}\big(\boldsymbol{z}^{(k)} - \boldsymbol{z}^{(k-1)}\big) \tag{57}$$

respectively. Because both the primal and dual residual must be zero for $\boldsymbol{x}^{(k)}$, $\boldsymbol{z}^{(k)}$, and $\boldsymbol{y}^{(k)}$ to be optimal [5, §3.3], and because increasing ρ tends to decrease the primal residual at the expense of increasing the dual residual (and vice-versa for a decrease in ρ) the idea is to balance their norms using

ρ:

$$\rho^{(k+1)} = \begin{cases} \tau^{\text{incr}}\rho^{(k)} & \text{if } \|\boldsymbol{r}^{(k+1)}\| > \mu\|\boldsymbol{s}^{(k+1)}\| \\ \rho^{(k)}/\tau^{\text{decr}} & \text{if } \|\boldsymbol{s}^{(k+1)}\| > \mu\|\boldsymbol{r}^{(k+1)}\| \\ \rho^{(k)} & \text{otherwise ,} \end{cases} \tag{58}$$

for constant $\tau^{\text{incr}}, \tau^{\text{decr}}, \mu \in \mathbb{R}$.

Translating into the language of our affine fixed point framework and using the same arguments as in (36) and (38),

$$\boldsymbol{r}^{(k+1)} = (\boldsymbol{y}^{(k+1)} - \boldsymbol{y}^{(k)})/\rho \approx \boldsymbol{v}_\rho/\rho \tag{59}$$

and

$$\boldsymbol{s}^{(k+1)} \approx -\rho \boldsymbol{A}^T \mathbb{G} \boldsymbol{v}_\rho \ . \tag{60}$$

We can therefore interpret residual balancing as comparing $\|\boldsymbol{v}_\rho\|$ with $\rho^2\|\boldsymbol{A}^T\mathbb{G}\boldsymbol{v}_\rho\|$, whereas our method compares $\|\boldsymbol{v}_\rho\|$ with $\rho\|\mathbb{G}\boldsymbol{v}_\rho\|$. The term $\rho^2\|\boldsymbol{A}^T\mathbb{G}\boldsymbol{v}_\rho\|$ does not appear in the eigenvector expression for the fixed point matrix $\boldsymbol{H}_\rho$, (33). However, we can view $\rho^2\|\boldsymbol{A}^T\mathbb{G}\boldsymbol{v}_\rho\|$ as a rough approximation of the term $\rho^2\|\mathbb{F}\mathbb{G}\boldsymbol{v}_\rho\| = \rho^2\|\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^T\mathbb{G}\boldsymbol{v}_\rho\|$, with equality when $\boldsymbol{A}$ and $\boldsymbol{Q}$ are orthogonal matrices. Taking this perspective, comparing $\|\boldsymbol{v}_\rho\|$ to $\rho^2\|\mathbb{F}\mathbb{G}\boldsymbol{v}_\rho\|$ is an alternative route to deriving the ρ small or ρ large approximations from Section III-A, and residual balancing may therefore be viewed as determining whether the ρ small or ρ large approximations hold by approximating $\rho^2\|\mathbb{F}\mathbb{G}\boldsymbol{v}_\rho\|$.

Our method is distinct from residual balancing because it uses a different approach to determining when ρ is too large or too small. It turns out (see Section V) that the approximation used by residual balancing has a significant theoretical disadvantage.

B. Barzilai-Borwein Spectral Method

The Barzilai-Borwein spectral (BBS) penalty parameter selection method [15], [16] involves rewriting ADMM as DRS applied to the dual of problem (1), rearranging DRS so that it resembles gradient descent on two variables, applying the Barzilai-Borwein method to choose the step sizes, and then translating back into the ADMM problem to select the penalty parameter ρ. We now interpret this method within our framework and explain the differences with the proposed method in detail.

Our notation	BBS [15]	SRB [17]
f, g	H, G	$\varphi,\ \psi$
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{c}$	A, B, b	$D,\ E,\ c$
$\boldsymbol{x}^{(k)}, \boldsymbol{z}^{(k)},$	u_k, v_k	$u^n,\ v^n$
$\boldsymbol{y}^{(k)}$	$-\lambda_k$	$-w^n$
$\tilde{\boldsymbol{y}}^{(k)}$	$-\hat{\lambda}_k$	
$\rho^{(k)}$	$\tau^{(k)}$	t_{n-1}

TABLE 1: Translation between the notation used here and that of [15] and [17].

Table 1 presents a list of symbol equivalences between [15] and our formulation. We can show (see Appendix B) that the expression of ADMM as DRS on the dual in [15] results in the same iteration on $\boldsymbol{y}$ that we derived by working with optimality conditions of the ADMM steps.

The key difference between the BBS method of [15] and the proposed method lies in (16) in [15], which, translated to our notation, becomes

$$\boldsymbol{A}F(\tilde{\boldsymbol{y}}) = a_f\tilde{\boldsymbol{y}} + \boldsymbol{c}_f \qquad \boldsymbol{B}G(\boldsymbol{y}) = a_g\boldsymbol{y} + \boldsymbol{c}_g \ , \tag{61}$$

where $a_f, a_g \in \mathbb{R}$ and $\boldsymbol{c}_f, \boldsymbol{c}_g \in \mathbb{R}^P$. What does this assumption say about f and g? If we assume f and g are quadratic as in (19), we fulfil the conditions in (61) when $\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^T = a_f\boldsymbol{I}$ and $\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T = a_g\boldsymbol{I}$. One simple way for this to hold is if $\boldsymbol{A}\boldsymbol{A}^T = a_{\boldsymbol{A}}\boldsymbol{I}$, $\boldsymbol{B}\boldsymbol{B}^T = a_{\boldsymbol{B}}\boldsymbol{I}$, and $\boldsymbol{Q}$ and $\boldsymbol{R}$ are themselves scaled identities ($\boldsymbol{Q} = a_{\boldsymbol{Q}}\boldsymbol{I}$ and $\boldsymbol{R} = a_{\boldsymbol{R}}\boldsymbol{I}$) and therefore

$$\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^T = a_{\boldsymbol{Q}}\boldsymbol{A}\boldsymbol{A}^T = a_{\boldsymbol{Q}}a_{\boldsymbol{A}}\boldsymbol{I} = a_f\boldsymbol{I} \tag{62}$$

and likewise for $\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T$.

Given assumption (16) in [15], our fixed point analysis provides a new route to derive the BBS selection rule of [15]. Following the quadratic analysis of Section II-B with the additional assumption from [15], we have

$$\mathbb{F} = a_f\boldsymbol{I} \qquad \mathbb{G} = a_g\boldsymbol{I} \ . \tag{63}$$

It follows that

$$\boldsymbol{H}_\rho = \frac{1 + \rho^2 a_f a_g}{(1+\rho a_f)(1+\rho a_g)}\boldsymbol{I} \ , \tag{64}$$

making it feasible to solve for the ρ that minimizes the leading constant, which is also equal to the spectral norm of $\boldsymbol{H}_\rho$. We have

$$\lambda(\rho) = \frac{1+\rho^2 a_f a_g}{(1+\rho a_f)(1+\rho a_g)} \tag{65}$$

$$\frac{d}{d\rho}\lambda(\rho) = \frac{(a_f + a_g)(\rho^2 a_f a_g - 1)}{(1+\rho a_f)^2(1+\rho a_g)^2} \ , \tag{66}$$

and therefore the minimum occurs when the numerator is zero, at

$$\rho = (a_f a_g)^{-\frac{1}{2}} \ , \tag{67}$$

which agrees with [15, Proposition 1].

What remains is to estimate a_f and a_g. The basic idea is that if $\mathbb{F}$ is a scaled identity, then, for arbitrary $\boldsymbol{w}$, we have

$$\frac{\boldsymbol{w}^T\mathbb{F}\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}} = a_f\frac{\boldsymbol{w}^T\boldsymbol{I}\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}} = a_f \ , \tag{68}$$

which provides a way to compute a_f by applying $\mathbb{F}$ to any vector $\boldsymbol{w}$. The challenge is computing this estimate from quantities that are readily available during the ADMM iterations. Recall that

$$\boldsymbol{A}F(\boldsymbol{y}) = -\mathbb{F}\boldsymbol{y} + C \qquad \boldsymbol{B}G(\boldsymbol{y}) = -\mathbb{G}\boldsymbol{y} + C \ , \tag{69}$$

which means that

$$\begin{aligned} \boldsymbol{A}F\big(\tilde{\boldsymbol{y}}^{(k)}\big) - \boldsymbol{A}F\big(\tilde{\boldsymbol{y}}^{(k-k_0)}\big) &= \boldsymbol{A}\boldsymbol{x}^{(k)} - \boldsymbol{A}\boldsymbol{x}^{(k-k_0)} \\ &= -\mathbb{F}\big(\tilde{\boldsymbol{y}}^{(k)} - \tilde{\boldsymbol{y}}^{(k-k_0)}\big) \ . \end{aligned} \tag{70}$$

Thus we can approximate a_f using

$$a_f \approx -\frac{\langle \boldsymbol{A}\Delta\boldsymbol{x}^{(k)}, \Delta\tilde{\boldsymbol{y}}^{(k)}\rangle}{\langle \Delta\tilde{\boldsymbol{y}}^{(k)}, \Delta\tilde{\boldsymbol{y}}^{(k)}\rangle} , \quad (71)$$

where $\Delta\tilde{\boldsymbol{y}}^{(k)} = \tilde{\boldsymbol{y}}^{(k)} - \tilde{\boldsymbol{y}}^{(k-\Delta k)}$ and $\Delta\boldsymbol{x} = \boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-\Delta k)}$. This agrees with the left side of (26) in [15] (note that (26) defines $\hat{\alpha} = 1/\alpha$).

A different estimate of a_f (the right side of (26) in [15]) may be obtained using (again for arbitrary $\boldsymbol{w}$)

$$\frac{(\mathbb{F}\boldsymbol{w})^T\mathbb{F}\boldsymbol{w}}{\boldsymbol{w}^T\mathbb{F}\boldsymbol{w}} = \frac{a_f^2}{a_f}\frac{\boldsymbol{w}^T\boldsymbol{I}\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{I}\boldsymbol{w}} = a_f , \quad (72)$$

which leads to

$$a_f \approx -\frac{\langle \boldsymbol{A}\Delta\boldsymbol{x}^{(k)}, \boldsymbol{A}\Delta\boldsymbol{x}^{(k)}\rangle}{\langle \boldsymbol{A}\Delta\boldsymbol{x}^{(k)}, \Delta\tilde{\boldsymbol{y}}^{(k)}\rangle} . \quad (73)$$

Using the same approach for $\mathbb{G}$ gives

$$a_g \approx -\frac{\langle \boldsymbol{B}\Delta\boldsymbol{z}^{(k)}, \Delta\boldsymbol{y}^{(k)}\rangle}{\langle \Delta\boldsymbol{y}^{(k)}, \Delta\boldsymbol{y}^{(k)}\rangle} \quad (74)$$

and

$$a_g \approx -\frac{\langle \boldsymbol{B}\Delta\boldsymbol{z}^{(k)}, \boldsymbol{B}\Delta\boldsymbol{z}^{(k)}\rangle}{\langle \boldsymbol{B}\Delta\boldsymbol{z}^{(k)}, \Delta\boldsymbol{y}^{(k)}\rangle} , \quad (75)$$

which are the reciprocals of the expressions after (28) in [15]. Substituting our expressions for a_f and a_g into (67) gives

$$\rho_{\text{BBS}}^{(k)} = \sqrt{\frac{\left\|\Delta\tilde{\boldsymbol{y}}^{(k)}\right\|^2 \left\|\Delta\boldsymbol{y}^{(k)}\right\|^2}{\langle \boldsymbol{A}(\Delta\boldsymbol{x}^{(k)}), \Delta\tilde{\boldsymbol{y}}^{(k)}\rangle \, \langle \boldsymbol{B}(\Delta\boldsymbol{z}^{(k)}), \Delta\boldsymbol{y}^{(k)}\rangle}} . \quad (76)$$

In summary, within the framework developed in this paper, one may view the BBS parameter selection method of [15] as being based on the minimization of the spectral radius of the linear fixed point iteration matrix $\boldsymbol{H}_\rho$ defined in (22) by making the additional assumption that $\mathbb{F}$ and $\mathbb{G}$ are (locally in the region around $\boldsymbol{y}^{(k)}$ and $\tilde{\boldsymbol{y}}^{(k)}$) scaled identity matrices. Under this assumption, the dependence of the spectral radius of $\boldsymbol{H}_\rho$ on ρ is simple and ρ may be selected to minimize the spectral radius of $\boldsymbol{H}_\rho$. The eigenvalues of $\mathbb{F}$ and $\mathbb{G}$ may be estimated by treating $\boldsymbol{y}^{(k+\Delta k)} - \boldsymbol{y}^{(k)}$ and $\tilde{\boldsymbol{y}}^{(k+\Delta k)} - \tilde{\boldsymbol{y}}^{(k)}$ as their eigenvectors, which is sensible under the local scaled identity assumption. This is a distinct view from that in [15], which is based on selecting a BBS step for the DRS algorithm and translating it into ADMM terms. We discuss further implementation details of this method in Section VI-A.

C. Spectral Radius Bound Method

The method of [17], which we refer to as the spectral radius bound (SRB) method, was derived as a step size selection method for DRS and then translated into the terminology of ADMM. The main idea is to minimize an upper bound on the spectral radius of the affine iteration matrix. We now discuss the approach in detail.

Table 1 provides a list of symbol equivalences between [17] and our formulation. It is clear by inspection that the fixed point mapping H_t from (6) in [17] is the same as $\boldsymbol{H}_\rho$ defined in (22). Thus [17, Lemma 2.1] (translated into our notation) states that, for eigenvector $\boldsymbol{v}$ with corresponding eigenvalue λ of $\boldsymbol{H}_\rho$, and assuming $\lambda \neq 1$, we have

$$\left|\lambda - \frac{1}{2}\right| \leq \sqrt{\frac{1}{4} - \frac{c}{1+2c}} \leq \frac{1}{2} , \quad (77)$$

where

$$c = \frac{\mathrm{Re}(\langle \mathbb{G}\boldsymbol{v}, \boldsymbol{v}\rangle)}{\rho^{-1}\|\boldsymbol{v}\|^2 + \rho\|\mathbb{G}\boldsymbol{v}\|^2} . \quad (78)$$

The SRB method uses the following heuristic to derive a way to select ρ from this bound: To force λ close to $1/2$, $c/(1+2c)$ should be as large as possible and therefore c should be as large as possible. For a fixed $\boldsymbol{v}$, this can be achieved by setting $\rho = \|\boldsymbol{v}\|/\|\mathbb{G}\boldsymbol{v}\|$.

To arrive at an implementable penalty parameter selection rule, it is heuristically assumed that $\boldsymbol{v} = \boldsymbol{y}$, i.e., that $\boldsymbol{y}$ is an eigenvector of $\boldsymbol{H}_\rho$. Together with definition $\mathbb{G}\boldsymbol{y} = \boldsymbol{B}\boldsymbol{z}$ this assumption results in

$$\rho_{\text{SRB}}^{(k)} = \frac{\left\|\boldsymbol{y}^{(k)}\right\|}{\left\|\boldsymbol{B}\boldsymbol{z}^{(k)}\right\|} . \quad (79)$$

This is somewhat ad hoc because it is not argued why $\boldsymbol{y}$ should be an eigenvector of $\boldsymbol{H}_\rho$, and because it ignores the dependence of $\boldsymbol{v}$ on ρ.

Of the approaches considered here, the SRB method [17] is the most conceptually similar to the proposed method. Both involve analyzing the spectral radius of the affine fixed point matrix that arises when ADMM is applied to a quadratic problem (although [17] does this in a roundabout way by instead analyzing the DRS algorithm and translating the results to ADMM). However, where [17] attempts to minimize the spectral radius by minimizing a bound on it, we instead approximate the spectral radius and avoid situations where ρ is clearly too large or too small. Both of these approaches involve comparing $\|\boldsymbol{v}\|$ with $\rho\|\mathbb{G}\boldsymbol{v}\|$, but this ratio is arrived at via different paths of reasoning. Finally, while the rules for selecting ρ are similar, (c.f. (55) and (79)) the proposed rule involves $\boldsymbol{y}^{(k+1)} - \boldsymbol{y}^{(k)}$ rather than $\boldsymbol{y}^{(k)}$ because we argue that this difference in $\boldsymbol{y}$ values should approximate a maximal eigenvector of $\boldsymbol{H}_\rho$. As described in Section V-B, it turns out that this difference results in a significant theoretical disadvantage for the SRB method.

V. Problem Transformations

The notion of problem transformations, originally identified in [13], provides a useful theoretical tool for comparing penalty parameter selection methods. The goal of this analysis is to identify how arbitrary decisions made during problem formulation, such as the choice of units for $\boldsymbol{x}$ and $\boldsymbol{z}$, affect the convergence of ADMM. A good penalty parameter selection method should be covariant or invariant to these transformations in the sense that, if it selects the optimal parameter for one problem, it should also select the corresponding optimal parameter for a transformed version of that problem. Here, we extend the scaling transform from [13] to include an additional degree of freedom in scaling, and include an additional transform based on translation of problem variables, which, to the best of our knowledge, has not previously been addressed in the literature.

A. Problem Scaling

Consider using ADMM to solve members of a family of optimization problems of the form

$$\operatorname*{arg\,min}_{\boldsymbol{x},\boldsymbol{z}} \alpha f(\gamma \boldsymbol{x}) + \alpha g(\delta \boldsymbol{z}) \ \text{ s.t. } \ \beta \boldsymbol{A} \gamma \boldsymbol{x} + \beta \boldsymbol{B} \delta \boldsymbol{z} = \beta \boldsymbol{c} \ , \tag{80}$$

where the family is parameterized by the scalars α, β, γ, and δ. We will refer to the problem with $\alpha = \beta = \gamma = \delta = 1$ as the *unscaled* problem, but it is important to emphasize that there is nothing special about this choice: the primary point of this analysis is that setting up a problem in ADMM form involves *implicit* choices of unknown values for these scalars, and *not* that it is useful to make *explicit* choices of these scalars to convert from one problem form to an equivalent one.

Denoting the solution to the unscaled problem by $\boldsymbol{x}^*$, $\boldsymbol{z}^*$, the solution to the problem with scaling α, β, γ, and δ is

$$\bar{\boldsymbol{x}}^* = \boldsymbol{x}^*/\gamma \qquad \bar{\boldsymbol{z}}^* = \boldsymbol{z}^*/\delta \ , \tag{81}$$

which may be verified by noting that α and β do not affect the minimizers of (80) and that γ and δ simply rescale $\boldsymbol{x}$ and $\boldsymbol{z}$. If we denote a particular choice of initialization for the unscaled problem as $\boldsymbol{z}^{(0)}$, $\boldsymbol{y}^{(0)}$, and ρ, how can we choose the initialization for a scaled problem, $\bar{\boldsymbol{z}}^{(0)}$, $\bar{\boldsymbol{y}}^{(0)}$, and $\bar{\rho}$, so that the sequences of variables generated by ADMM are properly scaled, i.e., so that $\bar{\boldsymbol{x}}^{(k)} = \boldsymbol{x}^{(k)}/\gamma$, and $\bar{\boldsymbol{z}}^{(k)} = \boldsymbol{z}^{(k)}/\delta$? The solution is provided by the initialization

$$\bar{\boldsymbol{z}}^{(0)} = \boldsymbol{z}^{(0)}/\delta \qquad \bar{\boldsymbol{y}}^{(0)} = \alpha \boldsymbol{y}^{(0)}/\beta \qquad \bar{\rho} = \alpha\rho/\beta^2 \ , \tag{82}$$

which may be verified via induction on the ADMM iterations [13, §III].

If we instead consider the adaptive version of ADMM where ρ may change after every iteration, we require $\bar{\rho}^{(k)} = \alpha\rho^{(k)}/\beta^2$, which motivates the following definition.

Definition 5.1 (Scaling Covariant):
An ADMM penalty parameter selection method, ϕ, is *scaling covariant* if

$$\phi\left(\left(\alpha\rho^{(j)}/\beta^2, \boldsymbol{x}^{(j+1)}/\gamma, \boldsymbol{z}^{(j+1)}/\delta, \alpha \boldsymbol{y}^{(j+1)}/\beta\right)_{j=0}^{k}\right) = \frac{\alpha}{\beta^2}\phi\left(\left(\rho^{(j)}, \boldsymbol{x}^{(j+1)}, \boldsymbol{z}^{(j+1)}, \boldsymbol{y}^{(j+1)}\right)_{j=0}^{k}\right) \ . \tag{83}$$

That is, if a method selects $\rho^{(k)}$ at iteration k of the unscaled problem, it should select $\alpha\rho^{(k)}/\beta^2$ for each corresponding scaled problem.

Parameter selection methods being scaling covariant is critical because problem scaling is unavoidable in practice. For example, for an inverse problem in imaging, the f term in (1) would typically represent a data fidelity functional involving a system model and the measurement vector. Because the measurement vector comes from the detector, its scaling is arbitrary, e.g., the manufacturer has converted a voltage to some physical units; we may convert again during preprocessing or scale the values to a convenient range (e.g., from 16-bit integer to floating point between zero and one). We also have freedom to choose the units for $\boldsymbol{x}$, e.g., one group may reconstruct in g/cm^3 and another in kg/m^3, as long as each implements their system model in a way that conforms to their choice of units for $\boldsymbol{x}$ and the measurements. Finally, the units in which we represent the error are arbitrary, e.g., sum of squares versus mean of squares. A similar argument can be made for the g term in (1), which would typically be the result of variable splitting applied to a regularization functional, $g(\boldsymbol{A}\boldsymbol{x})$. Again, the units of $\boldsymbol{x}$ are arbitrary, as is the implementation of $\boldsymbol{A}$ (e.g., finite differences may be unscaled, divided by two, or divided by the physical pixel spacing). The units for the output of g are similarly arbitrary, but changing them affects the weighting of the regularization term relative to the data fidelity. If the relative weighting is assumed fixed, i.e., it is always properly tuned, the scaling of the output of g is fixed (hence a single scaling parameter, α, on f and g in (80) rather than one for each). Finally, similar scaling arguments may be made for $\boldsymbol{B}$ or $\boldsymbol{c}$.

B. Problem Translation

In addition to freedom in problem scaling, a problem statement of the form (1) admits freedom in terms of variable translation. Like scaling, these translations can be seen as a consequence of choosing units for $\boldsymbol{x}$ and $\boldsymbol{z}$, e.g., if $\boldsymbol{x}$ represents temperature, it may be represented in degrees Celsius or Kelvin.

Consider the problem

$$\begin{aligned} &\operatorname*{arg\,min}_{\boldsymbol{x},\boldsymbol{z}} f(\boldsymbol{x} + \boldsymbol{x}_0) + g(\boldsymbol{z} + \boldsymbol{z}_0) \\ &\text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{c} - \boldsymbol{A}\boldsymbol{x}_0 - \boldsymbol{B}\boldsymbol{z}_0 \ . \end{aligned} \tag{84}$$

We now show that with proper choice of initialization, ADMM applied to the translated problem (84) results in a translated sequence of iterates (and solution) as compared to the untranslated version. Let $\bar{\boldsymbol{x}}^{(k)}$, $\bar{\boldsymbol{z}}^{(k)}$, $\bar{\boldsymbol{y}}^{(k)}$, and $\bar{\rho}^{(k)}$ denote ADMM variables for the translated problem, and define $\bar{c} = \boldsymbol{c} - \boldsymbol{A}\boldsymbol{x}_0 - \boldsymbol{B}\boldsymbol{z}_0$. Assume for purposes of induction that

$$\bar{\boldsymbol{z}}^{(k)} = \boldsymbol{z}^{(k)} - \boldsymbol{z}_0 \qquad \bar{\boldsymbol{y}}^{(k)} = \boldsymbol{y}^{(k)} \qquad \bar{\rho}^{(k)} = \rho^{(k)} \ . \tag{85}$$

We have

$$\begin{aligned} &\frac{\bar{\rho}^{(k)}}{2}\left\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\bar{\boldsymbol{z}}^{(k)} - \bar{\boldsymbol{c}} + \frac{1}{\bar{\rho}^{(k)}}\bar{\boldsymbol{y}}^{(k)}\right\|^2 \\ &= \frac{\rho^{(k)}}{2}\left\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\left(\boldsymbol{z}^{(k)} - \boldsymbol{z}_0\right) - \boldsymbol{c} + \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{B}\boldsymbol{z}_0 + \frac{1}{\rho^{(k)}}\boldsymbol{y}^{(k)}\right\|^2 \\ &= \frac{\rho^{(k)}}{2}\left\|\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{x}_0) + \boldsymbol{B}\boldsymbol{z}^{(k)} - \boldsymbol{c} + \frac{1}{\rho^{(k)}}\boldsymbol{y}^{(k)}\right\|^2 \end{aligned} \tag{86}$$

and therefore

$$\bar{\boldsymbol{x}}^{(k+1)} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}_0 \ . \tag{87}$$

A similar argument can be used to show that $\bar{\boldsymbol{x}}^{(k)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}_0$ implies that $\bar{\boldsymbol{z}}^{(k+1)} = \boldsymbol{z}^{(k+1)} - \boldsymbol{z}_0$. Setting $\bar{\boldsymbol{x}}^{(0)} = \boldsymbol{x}^{(0)} - \boldsymbol{x}_0$ and $\bar{\boldsymbol{z}}^{(0)} = \boldsymbol{z}^{(0)} - \boldsymbol{z}_0$ completes the induction. Finally, a change of variables shows that if $(\boldsymbol{x}^*, \boldsymbol{z}^*)$ is a solution to problem (1), then $(\boldsymbol{x}^* - \boldsymbol{x}_0, \boldsymbol{z}^* - \boldsymbol{z}_0)$ is a solution to the translated problem (84). These relationships motivate the following definition.

Definition 5.2 (Translation Invariant):
An ADMM penalty parameter selection method, ϕ, is *translation invariant* if

$$\phi\left(\left(\rho^{(j)}, \boldsymbol{x}^{(j+1)}, \boldsymbol{z}^{(j+1)}, \boldsymbol{y}^{(j+1)}\right)_{j=0}^{k}\right) = \phi\left(\left(\rho^{(j)}, \boldsymbol{x}^{(j+1)} - \boldsymbol{x}_0, \boldsymbol{z}^{(j+1)} - \boldsymbol{z}_0, \boldsymbol{y}^{(j+1)}\right)_{j=0}^{k}\right) . \tag{88}$$

That is, if a method selects $\rho^{(k)}$ at iteration k of the untranslated problem, it should still select $\rho^{(k)}$ for each corresponding translated problem.

C. Effects of Problem Transformations on Parameter Selection

Penalty parameter selection methods should be both scaling covariant and translation invariant. If a method is not, then even if it can select the optimal parameter for one scaling/translation, it will select a suboptimal one for a different scaling/translation. Stated differently, the resulting convergence performance will be dependent on the arbitrary choices made during problem specification. We now discuss how each of the three exiting methods described in Section IV and the proposed method from Section III perform under problem transformation.

1) Residual Balancing

As demonstrated in [13], the residual balancing method (Section IV-A) is not scaling covariant. For residual balancing as defined in (58) to satisfy the scaling covariance property (83), we need the ratio of the norms of the primal and dual residuals to be scaling invariant. Instead, we have

$$\begin{aligned}\bar{\boldsymbol{r}}^{(k)} &= \beta \boldsymbol{A}\gamma\bar{\boldsymbol{x}}^{(k)} + \beta\boldsymbol{B}\delta\bar{\boldsymbol{z}}^{(k)} - \beta\boldsymbol{c} \\ &= \beta\boldsymbol{A}\boldsymbol{x}^{(k)} + \beta\boldsymbol{B}\boldsymbol{z}^{(k)} - \beta\boldsymbol{c} \\ &= \beta\boldsymbol{r}^{(k)}\end{aligned} \tag{89}$$

and

$$\begin{aligned}\bar{\boldsymbol{s}}^{(k)} &= \bar{\rho}^{(k)}\beta^2\gamma\delta\boldsymbol{A}^T\boldsymbol{B}\big(\bar{\boldsymbol{z}}^{(k)} - \bar{\boldsymbol{z}}^{(k-1)}\big) \\ &= \bar{\rho}^{(k)}\beta^2\gamma\boldsymbol{A}^T\boldsymbol{B}\big(\boldsymbol{z}^{(k)} - \boldsymbol{z}^{(k-1)}\big) \\ &= \alpha\rho^{(k)}\gamma\boldsymbol{A}^T\boldsymbol{B}\big(\boldsymbol{z}^{(k)} - \boldsymbol{z}^{(k-1)}\big) \\ &= \alpha\gamma\boldsymbol{s}^{(k)} .\end{aligned} \tag{90}$$

Because $\|\bar{\boldsymbol{r}}\|/\|\bar{\boldsymbol{s}}\| \neq \|\boldsymbol{r}\|/\|\boldsymbol{s}\|$, the choice of how to change ρ from (58) depends on problem scaling.

The residual balancing method is translation invariant. We have

$$\begin{aligned}\bar{\boldsymbol{r}}^{(k)} &= \boldsymbol{A}\bar{\boldsymbol{x}}^{(k)} + \boldsymbol{B}\bar{\boldsymbol{z}}^{(k)} - \bar{\boldsymbol{c}} \\ &= \boldsymbol{A}(\boldsymbol{x}^{(k)} - \boldsymbol{x}_0) + \boldsymbol{B}\big(\boldsymbol{z}^{(k)} - \boldsymbol{z}_0\big) - \boldsymbol{c} + \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{B}\boldsymbol{z}_0 \\ &= \boldsymbol{A}\boldsymbol{x}^{(k)} + \boldsymbol{B}\boldsymbol{z}^{(k)} - \boldsymbol{c} \\ &= \boldsymbol{r}^{(k)}\end{aligned} \tag{91}$$

and

$$\begin{aligned}\bar{\boldsymbol{s}}^{(k)} &= -\bar{\rho}^{(k)}\boldsymbol{A}^T\boldsymbol{B}\big(\bar{\boldsymbol{z}}^{(k)} - \bar{\boldsymbol{z}}^{(k-1)}\big) \\ &= -\rho^{(k)}\boldsymbol{A}^T\boldsymbol{B}\big(\boldsymbol{z}^{(k)} - \boldsymbol{z}^{(k-1)}\big) \\ &= \boldsymbol{s}^{(k)} ,\end{aligned} \tag{92}$$

which means that, if initialized correctly, residual balancing will choose the same $\rho^{(k)}$ sequence no matter how a problem is translated.

2) Barzilai-Borwein Spectral Method

The BBS method (Section IV-B) is scaling covariant because

$$\bar{a}_f^{(k)} = -\frac{\langle\beta\boldsymbol{A}\gamma\Delta\bar{\boldsymbol{x}}^{(k)}, \beta\boldsymbol{A}\gamma\Delta\bar{\boldsymbol{x}}^{(k)}\rangle}{\langle\beta\boldsymbol{A}\gamma\Delta\bar{\boldsymbol{x}}^{(k)}, \Delta\bar{\tilde{\boldsymbol{y}}}^{(k)}\rangle} \tag{93}$$

$$= -\frac{\beta\langle\boldsymbol{A}\Delta\boldsymbol{x}^{(k)}, \boldsymbol{A}\Delta\boldsymbol{x}^{(k)}\rangle}{\langle\boldsymbol{A}\Delta\boldsymbol{x}^{(k)}, \Delta\alpha\tilde{\boldsymbol{y}}^{(k)}/\beta\rangle} \tag{94}$$

$$= \frac{\beta^2}{\alpha}a_f^{(k)} , \tag{95}$$

and likewise

$$\bar{a}_g^{(k)} = \frac{\beta^2}{\alpha}a_g^{(k)} . \tag{96}$$

Therefore

$$\bar{\rho}^{(k)} = \big(\bar{a}_f^{(k)}\bar{a}_g^{(k)}\big)^{-\frac{1}{2}} = \frac{\alpha}{\beta^2}\rho^{(k)} , \tag{97}$$

which is the scaling required to make the scaled problem converge in the same way as the standardized one.

The BBS method is translation invariant because the expression for ρ only involves quantities that are differences between variables at different iterations.

3) Spectral Radius Bound Method

The SRB method (Section IV-C) is scaling covariant because it sets

$$\bar{\rho}^{(k)} = \frac{\big\|\bar{\boldsymbol{y}}^{(k)}\big\|}{\big\|\beta\boldsymbol{B}\delta\bar{\boldsymbol{z}}^{(k)}\big\|} = \frac{\big\|\alpha\boldsymbol{y}^{(k)}/\beta\big\|}{\big\|\beta\boldsymbol{B}\boldsymbol{z}^{(k)}\big\|} = \frac{\alpha}{\beta^2}\rho^{(k)} , \tag{98}$$

which is the scaling required to make the scaled problem converge in the same way as the standardized one.

The SRB method is not translation invariant, because

$$\bar{\rho}^{(k)} = \frac{\big\|\bar{\boldsymbol{y}}^{(k)}\big\|}{\big\|\boldsymbol{B}\bar{\boldsymbol{z}}^{(k)}\big\|} = \frac{\big\|\boldsymbol{y}^{(k)}\big\|}{\big\|\boldsymbol{B}(\boldsymbol{z}^{(k)} - \boldsymbol{z}_0)\big\|} \neq \rho^{(k)} . \tag{99}$$

As a result, we expect that when the SRB method provides good convergence on one problem, it may not provide good convergence on a translated version of that problem.

4) Proposed Method

The proposed method (Section III) is scaling covariant because

$$\begin{aligned}\frac{\bar{\rho}^{(k)}\left\|\beta \boldsymbol{B}\delta\left(\bar{\boldsymbol{z}}^{(k+1)}-\bar{\boldsymbol{z}}^{(k)}\right)\right\|}{\left\|\bar{\boldsymbol{y}}^{(k+1)}-\bar{\boldsymbol{y}}^{(k)}\right\|} &= \frac{\frac{\alpha}{\beta^2}\rho^{(k)}\left\|\beta \boldsymbol{B}\left(\boldsymbol{z}^{(k+1)}-\boldsymbol{z}^{(k)}\right)\right\|}{\frac{\alpha}{\beta}\left\|\boldsymbol{y}^{(k+1)}-\boldsymbol{y}^{(k)}\right\|}\\ &= \frac{\rho^{(k)}\left\|\boldsymbol{B}\left(\boldsymbol{z}^{(k+1)}-\boldsymbol{z}^{(k)}\right)\right\|}{\left\|\boldsymbol{y}^{(k+1)}-\boldsymbol{y}^{(k)}\right\|}\,.\end{aligned} \tag{100}$$

As a result, any decision about how to change ρ based on this ratio will be the same no matter the problem scaling.

The proposed ratio (Section III) is translation invariant, because, like the BBS method, it only involves differences between variables at different iterations.

VI. Computational Experiments

We now describe our experiments comparing the proposed method to standard, non-adaptive ADMM and three state-of-the-art adaptive ADMM approaches.

A. Implementation Details for Parameter Selection Methods

So far, we have focused on describing how various methods determine what the value of the penalty parameter ρ should be, or whether it is currently too large or small. In some of these cases there is more than one way of constructing a corresponding penalty parameter selection algorithm. Because a comprehensive study of these options is well beyond the scope of this paper, for each comparison method, we use the specific algorithm recommended in the paper that proposed it. We now briefly summarize these methods.

For the **residual balancing (RB) method** (Section IV-A), we followed the algorithm described in (58) with $\tau^{\text{incr}} = 2$, $\tau^{\text{decr}} = 2$, and $\mu = 10$ as suggested by [5].

For the **Barzilai-Borwein spectral (BBS) method** (Section IV-B), we followed the algorithm from [15, Algorithm 1] and the code provided by the authors.[8] This algorithm involves computing a_f and a_g as described in (68)-(75), combining those estimates, and using safeguarding rules that attempt to discard the estimates when underlying assumptions are not met. We used the recommended safeguarding parameter $\epsilon^{\text{cor}} = 0.2$ and update frequency $T_f = 2$.

For the **spectral radius bound (SRB) method** (Section IV-C), we followed the algorithm in [17], in which the current ρ and ρ_{SRB} (79) are mixed with a decaying weight on ρ_{SRB}. The result is also clipped so that it always falls within a user-defined range. We used a weight decay schedule of $2^{-k/100}$ and a range of $[10^{-4}, 10^4]$ as recommended in [17].

For the proposed **spectral radius approximation (SRA) method**, we considered several possible ways to turn the rule (55) into a penalty parameter selection method. How often should ρ be updated? Should $\rho^{(k+1)}$ be set to $\rho_{\text{SRA}}^{(k+1)}$, set to some combination of $\rho^{(k)}$ and $\rho_{\text{SRA}}^{(k+1)}$, or simply be moved in the direction of $\rho_{\text{SRA}}^{(k+1)}$? Should the update based on $\rho_{\text{SRA}}^{(k+1)}$ take into account that it is expected to be less reliable when $\|\rho\mathbb{G}\boldsymbol{v}_{\rho_0}\| \approx \|\boldsymbol{v}_{\rho_0}\|$? Should we constrain $\rho^{(k+1)}$ to lie within some interval determined by prior estimates? How should the $\rho_{\text{SRA}}^{(k+1)} = 0$ and $\rho_{\text{SRA}}^{(k+1)} = \infty$ cases be handled? Since a thorough exploration of these options would be a significant undertaking, we have deferred it to future work, implementing the simple approach in Algorithm 1. This algorithm sets ρ based on the ratio (55) except when it is 0 or ∞, in which case ρ is multiplied or divided by a fixed scalar (we used $\tau^{\text{incr}} = \tau^{\text{decr}} = 10$). It also only makes adjustments every few iterations (we used $T = 5$). The trade-off in the choice of T is that a large T makes the approximations involved in deriving the SRA rule more accurate (because they are asymptotic in k), while a small T means ρ is updated more frequently, which may accelerate convergence.

Algorithm 1: Proposed ρ selection method

```
Input: $k, \rho^{(k)}, \boldsymbol{z}^{(k)}, \boldsymbol{z}^{(k+1)}, \boldsymbol{y}^{(k)}, \boldsymbol{y}^{(k+1)}$
Parameters: $T = 5, \tau^{\text{incr}} = \tau^{\text{decr}} = 10$
Output: $\rho^{(k+1)}$
if $k \mod T \neq 1$ :  // update every T steps
    return $\rho^{(k)}$
$p \leftarrow \left\|\boldsymbol{y}^{(k+1)} - \boldsymbol{y}^{(k)}\right\|_2$            // (55) numerator
$q \leftarrow \left\|\boldsymbol{B}\left(\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)}\right)\right\|_2$  // (55) denominator
if $p = 0$ and $q > 0$ :                     // $p/q = 0$
    return $\rho^{(k)}/\tau^{\text{decr}}$
if $p > 0$ and $q = 0$ :                     // $p/q = \infty$
    return $\tau^{\text{incr}}\rho^{(k)}$
if $p = 0$ and $q = 0$ :                     // $p/q = 0/0$
    return $\rho^{(k)}$
return $p/q$
```

We also compared with not adapting the penalty parameter, i.e., $\rho^{(k)} = \rho^{(0)}$, which we refer to as the **fixed** method.

B. Sum of Quadratics

Our first experiment considered the sum of quadratics problem

$$\begin{aligned}&\operatorname*{arg\,min}_{\boldsymbol{x},\boldsymbol{z}} \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{q}^T\boldsymbol{x} + \frac{1}{2}\boldsymbol{z}^T\boldsymbol{R}\boldsymbol{z} + \boldsymbol{r}^T\boldsymbol{z},\\ &\qquad\qquad\text{s.t.}\quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{c}\,,\end{aligned} \tag{101}$$

with variables $\boldsymbol{x} \in \mathbb{R}^M$ and $\boldsymbol{z} \in \mathbb{R}^N$; vectors $\boldsymbol{q} \in \mathbb{R}^M$, $\boldsymbol{r} \in \mathbb{R}^N$, and $\boldsymbol{c} \in \mathbb{R}^P$; and matrices $\boldsymbol{Q} \in \mathbb{R}^{M\times M}$, $\boldsymbol{R} \in \mathbb{R}^{N\times N}$, $\boldsymbol{A} \in \mathbb{R}^{P\times M}$, and $\boldsymbol{B} \in \mathbb{R}^{P\times N}$.

While one would not typically solve (101) using ADMM, it represents an important reference experiment due to the fundamental role of quadratic approximations in the proposed framework (i.e., the approximations used to derive the framework hold exactly in this case), and because the

[8] Available from https://github.com/nightldj/admm_release .

solution can be computed to high precision (aiding in performance comparisons) via efficient problem-specific methods. The ADMM iterations for the quadratic problem are

$$\boldsymbol{x}^{(k+1)} = -(\boldsymbol{Q} + \rho \boldsymbol{A}^T \boldsymbol{A})^{-1} \left(\boldsymbol{q} - \rho \boldsymbol{A}^T \left(\boldsymbol{c} - \boldsymbol{B}\boldsymbol{z}^k - \frac{\boldsymbol{y}^{(k)}}{\rho} \right) \right) \quad (102)$$

$$\boldsymbol{z}^{(k+1)} = -(\boldsymbol{R} + \rho \boldsymbol{B}^T \boldsymbol{B})^{-1} \left(\boldsymbol{r} - \rho \boldsymbol{B}^T \left(\boldsymbol{c} - \boldsymbol{A}\boldsymbol{x}^{(k+1)} - \frac{\boldsymbol{y}^{(k)}}{\rho} \right) \right) \quad (103)$$

$$\boldsymbol{y}^{(k+1)} = \boldsymbol{y}^{(k)} + \rho\big(\boldsymbol{A}\boldsymbol{x}^{(k+1)} + \boldsymbol{B}\boldsymbol{z}^{(k+1)} - \boldsymbol{c}\big) \,. \quad (104)$$

We can find the solution without using ADMM by rewriting the problem as

$$\begin{aligned} \arg\min_{\boldsymbol{w}} \; & \frac{1}{2}\boldsymbol{w}^T \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix} \boldsymbol{w} + \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{r} \end{bmatrix}^T \boldsymbol{w} \\ \text{s.t.} \; & \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \end{bmatrix} \boldsymbol{w} = \boldsymbol{c} \,, \end{aligned} \quad (105)$$

where $\boldsymbol{w} = \begin{bmatrix} \boldsymbol{x}^T & \boldsymbol{z}^T \end{bmatrix}^T$ is the concatenation of the original optimization variables. We now have a quadratic problem with an affine constraint. Letting Φ denote an orthogonal basis for the null space of $\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \end{bmatrix}$ and $\boldsymbol{w}_0$ denote any vector such that $\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \end{bmatrix} \boldsymbol{w}_0 = \boldsymbol{c}$, we can instead solve for an optimal $\boldsymbol{\alpha}^*$ using

$$\begin{aligned} & \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}(\Phi\boldsymbol{\alpha} + \boldsymbol{w}_0)^T \boldsymbol{H} (\Phi\boldsymbol{\alpha} + \boldsymbol{w}_0) + \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{r} \end{bmatrix}^T (\Phi\boldsymbol{\alpha} + \boldsymbol{w}_0) \\ & = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T \Phi^T \boldsymbol{H} \Phi \boldsymbol{\alpha} + \left(\Phi^T \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{r} \end{bmatrix} + \Phi^T \boldsymbol{H} \boldsymbol{w}_0 \right)^T \boldsymbol{\alpha} \,, \end{aligned} \quad (106)$$

where

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix} . \quad (107)$$

The solution to the original problem is then given by $\begin{bmatrix} \boldsymbol{x}^{*T} & \boldsymbol{z}^{*T} \end{bmatrix}^T = \Phi\boldsymbol{\alpha}^* + \boldsymbol{w}_0$. This approach can be implemented efficiently[9] when the number of dimensions is in the hundreds.

We constructed an instance of problem (101) with $M = 15$, $N = 13$, and $P = 8$; with $\boldsymbol{q}$, $\boldsymbol{r}$, $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{c}$ generated with random normal entries; and with and $\boldsymbol{Q}$ and $\boldsymbol{R}$ separately constructed from a product $\boldsymbol{X}^T\boldsymbol{X}$, where $\boldsymbol{X}$ had random normal entries to ensure they were symmetric and positive semidefinite.

To validate our theoretical framework, we formed the affine iteration matrix $\boldsymbol{H}_\rho$ from (22) and computed its spectral radius numerically for a range of ρ values. The results in Figs. 1 and 2 demonstrate that the spectral radius can change in a complicated way and that the limiting behaviors developed in Section III-A are remarkably accurate.

We applied each of the five methods described in Section VI-A to this problem, with $\boldsymbol{x}^{(0)} = \boldsymbol{0}$, $\boldsymbol{z}^{(0)} = \boldsymbol{0}$, and $\boldsymbol{y}^{(0)} = \boldsymbol{0}$, while varying $\rho^{(0)}$ logarithmically in the range 10^{-3} to 10^3 with 5 values per decade. We then repeated this experiment on a scaled version of (101) with $\alpha = 10^3$ and a translated version with the translation $\boldsymbol{z}_0$ chosen as a Gaussian random vector with standard deviation 10.

For this experiment, we quantified the performance of each method by computing the relative residual between $\boldsymbol{x}^{(k)}$ and $\boldsymbol{x}^*$ at $k = 50$. (Whether convergence of the variables or convergence of the functional is more meaningful is application dependent. In developing these experiments, we did not observe this choice to affect trends in the results.)

Results are shown in Fig. 4. The results for a fixed penalty parameter show relative convergence varying by several orders of magnitude. As expected, if the optimal ρ is selected in advance and not adapted online, it leads to excellent convergence. Each of the penalty parameter selection methods mitigates this effect to some degree, improving convergence across different initializations of the penalty parameter. The residual balancing method is better than a fixed parameter for the basic quadratic problem and the translated version, but its performance is poor on the scaled problem, which is in line with the analysis of Section V showing that it is not scaling covariant. SRB performs well on the basic and scaled problem, but poorly on the translated one, also in line with Section V, which shows that it is not translation invariant. Both BBS and the proposed method are scaling covariant and translation invariant, but the proposed method is empirically superior, resulting in better convergence for a wide range of parameter initializations, which is presumably because the proposed method uses a more general quadratic approximation (see Section IV-B).

C. Basis Pursuit Denoising

Basis pursuit denoising (BPDN) [30], which finds a sparse representation of a signal or image in a fixed dictionary, is formulated as

$$\arg\min_{\boldsymbol{x}} \underbrace{\frac{1}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{d}\|_2^2 + w\|\boldsymbol{x}\|_1}_{J(\boldsymbol{x})} \,, \quad (108)$$

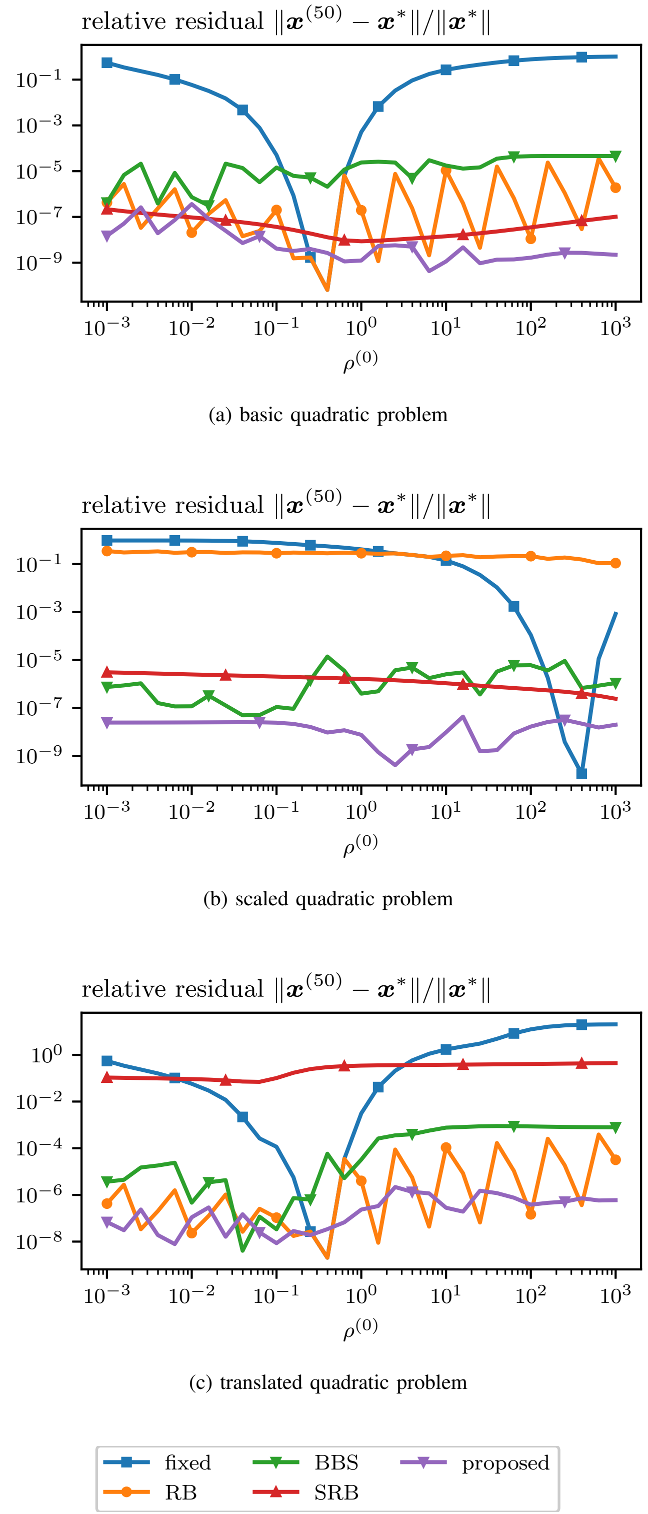
with dictionary matrix $\boldsymbol{D} \in \mathbb{R}^{K \times M}$, variable $\boldsymbol{x} \in \mathbb{R}^M$, fixed vector $\boldsymbol{d} \in \mathbb{R}^K$, scalar parameter $w \geq 0$, and where $\|\cdot\|_2$ denotes the ℓ_2 norm, $\|\cdot\|_1$ denotes the ℓ_1 norm, and $J : \mathbb{R}^M \to \mathbb{R}$ is the functional we aim to minimize. It can be expressed in the form of an ADMM problem (1) via variable splitting, resulting in

$$\arg\min_{\boldsymbol{x},\boldsymbol{z}} \frac{1}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{d}\|_2^2 + w\|\boldsymbol{z}\|_1 \quad \text{s.t.} \quad \boldsymbol{x} - \boldsymbol{z} = \boldsymbol{0} \,, \quad (109)$$

which corresponds to

$$\boldsymbol{A} = \boldsymbol{I} \qquad \boldsymbol{B} = -\boldsymbol{I} \qquad \boldsymbol{c} = \boldsymbol{0} \quad (110)$$

[9] For example, in Python, by using `scipy.linalg.null_space` and `scipy.linalg.solve`.



(a) basic quadratic problem

(b) scaled quadratic problem

(c) translated quadratic problem

FIGURE 4: ADMM penalty parameter methods evaluated in terms of convergence on sum of quadratics problems. The RB method is not scaling covariant and the SRB method is not translation invariant.

in (1). The ADMM iterations are

$$\boldsymbol{x}^{(k+1)} = (\boldsymbol{D}^T\boldsymbol{D} + \rho\boldsymbol{I})^{-1}\left(\boldsymbol{D}^T\boldsymbol{d} + \rho\left(\boldsymbol{z}^{(k)} - \frac{\boldsymbol{y}^{(k)}}{\rho}\right)\right) \tag{111}$$

$$\boldsymbol{z}^{(k+1)} = \mathcal{S}_{w/\rho}\left(\boldsymbol{x}^{(k+1)} + \frac{\boldsymbol{y}^{(k)}}{\rho}\right) \tag{112}$$

$$\boldsymbol{y}^{(k+1)} = \boldsymbol{y}^{(k)} + \rho\big(\boldsymbol{x}^{(k+1)} - \boldsymbol{z}^{(k+1)}\big)\ , \tag{113}$$

where $\mathcal{S}$ denotes the proximal operator of the ℓ_1 norm, also referred to as the soft thresholding operator [5, §4.4.3].

For our experiment, we used BPDN to solve the regression problem involving a diabetes dataset[10] that was addressed in [31] and that was also one of the example problems considered in [15] (although using the closely-related elastic net problem rather than BPDN). The dimension of the data $\boldsymbol{d}$ was $K = 442$ and the dimension of the sparse code $\boldsymbol{x}$ was $M = 10$. We solved the $\boldsymbol{x}$-update via LU factorization. We quantified performance by comparing the value of the objective functional at iteration 50, $J(\boldsymbol{x}^{(50)})$, to the minimal functional value obtained by any method when run for 100 total iterations, which we denote J^*. Results are shown in Fig. 5a and discussed in Section VI-H.

D. Robust PCA

Robust principal component analysis (robust PCA) [32] is a matrix decomposition technique based on solving the optimization problem

$$\operatorname*{arg\,min}_{\boldsymbol{X},\boldsymbol{Z}} \ \underbrace{\|\boldsymbol{X}\|_* + w\|\boldsymbol{Z}\|_1}_{J(\boldsymbol{X},\boldsymbol{Z})} \quad \text{s.t.} \quad \boldsymbol{X} + \boldsymbol{Z} = \boldsymbol{D}\ , \tag{114}$$

where $\|\cdot\|_*$ denotes the nuclear norm, $\boldsymbol{D}$ is the matrix to be decomposed, $\boldsymbol{X}$ is the low-rank component of the data, and $\boldsymbol{Z}$ is the sparse component of the data. The problem is already posed in standard ADMM form, with

$$\boldsymbol{A} = \boldsymbol{I} \qquad \boldsymbol{B} = \boldsymbol{I} \qquad \boldsymbol{C} = \boldsymbol{D} \tag{115}$$

in (1) (where the vector variables $\boldsymbol{x}$, $\boldsymbol{z}$, and $\boldsymbol{c}$ should be replaced with corresponding matrix variables $\boldsymbol{X}$, $\boldsymbol{Z}$, and $\boldsymbol{C}$ in this case). The ADMM iterations are

$$\boldsymbol{X}^{(k+1)} = \mathcal{T}_{1/\rho}\left(\boldsymbol{D} - \boldsymbol{Z}^{(k)} - \frac{\boldsymbol{Y}^{(k)}}{\rho}\right) \tag{116}$$

$$\boldsymbol{Z}^{(k+1)} = \mathcal{S}_{w/\rho}\left(\boldsymbol{D} - \boldsymbol{X}^{(k+1)} - \frac{\boldsymbol{Y}^{(k)}}{\rho}\right) \tag{117}$$

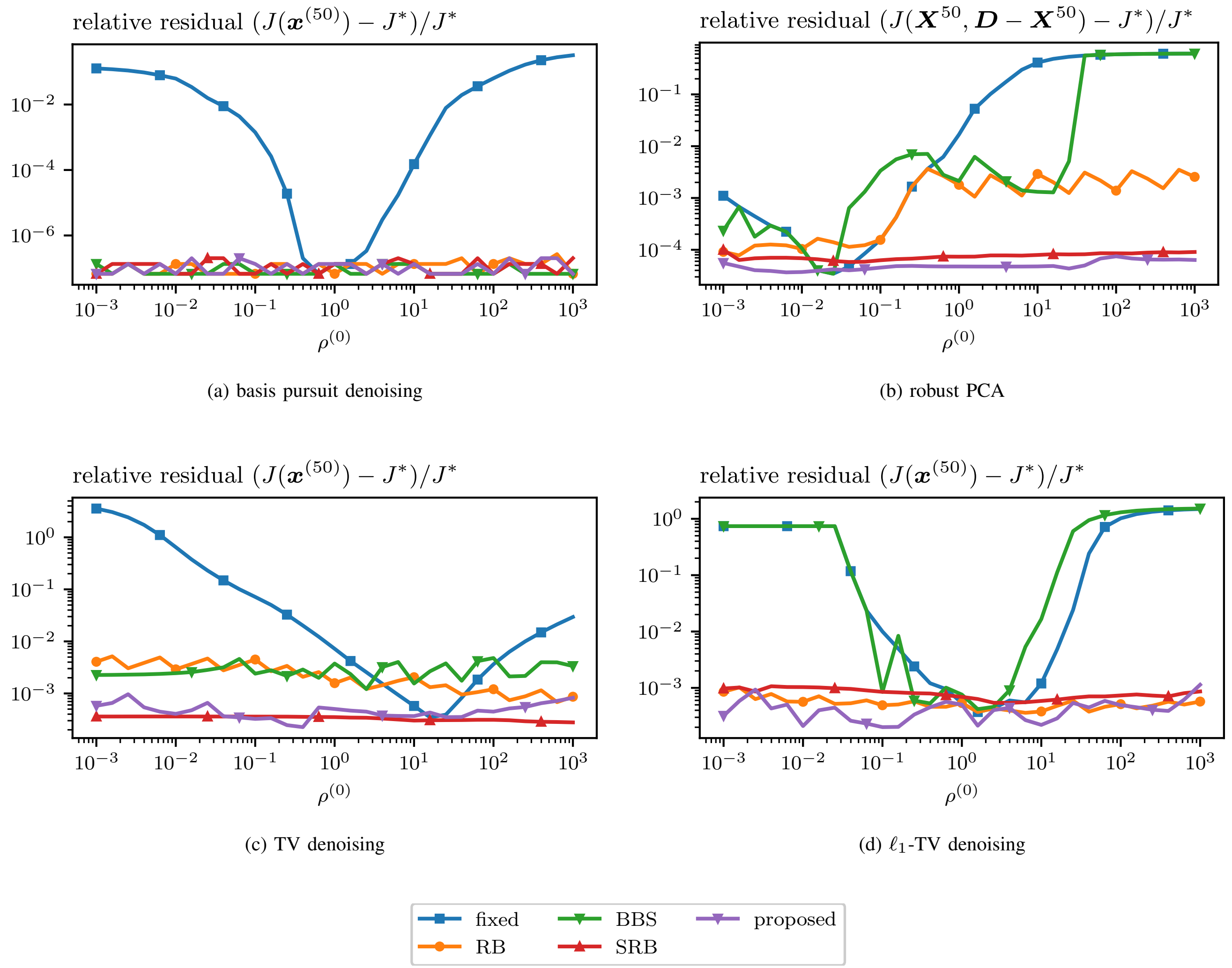
$$\boldsymbol{Y}^{(k+1)} = \boldsymbol{Y}^{(k)} + \rho\left(\boldsymbol{D} - \boldsymbol{X}^{(k+1)} - \boldsymbol{Z}^{(k+1)}\right)\ , \tag{118}$$

where $\mathcal{T}$ is the scaled proximal operator of the nuclear norm [33].

Our experiment addressed the video background/foreground separation problem, which is one of the many applications of this technique, using the Lankershim Boulevard traffic camera dataset.[11] To quantify

[10] Available from https://hastie.su.domains/Papers/LARS/ .

[11] Available from https://data.transportation.gov/Automobiles/Next-Generation-Simulation-NGSIM-Program-Lankershi/uv3e-y54k .



(a) basis pursuit denoising

(b) robust PCA

(c) TV denoising

(d) ℓ_1-TV denoising

FIGURE 5: ADMM penalty parameter method evaluation. Better methods give a lower relative residual across a wider range $\rho^{(0)}s$.

performance, we found a feasible $\boldsymbol{Z}$ by subtracting $\boldsymbol{X}$ from $\boldsymbol{D}$, computed the value of the objective functional, $J(\boldsymbol{X}^{(50)}, \boldsymbol{D} - \boldsymbol{X}^{(50)})$, and compared it to the best such value attained by any method after 100 iterations, J^*. Results are shown in Fig. 5b and discussed in Section VI-H.

E. TV Denoising

Total variation (TV) denoising can be expressed as the optimization problem

$$\arg\min_{\boldsymbol{x}} \underbrace{\frac{1}{2}\left\|\boldsymbol{x} - \boldsymbol{d}\right\|_2^2 + w\left\|\sqrt{(\boldsymbol{G}_0\boldsymbol{x})^2 + (\boldsymbol{G}_1\boldsymbol{x})^2}\right\|_1}_{J(\boldsymbol{x})}, \tag{119}$$

where $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$ are gradient operators along the first and second axis of the image $\boldsymbol{x}$. The ADMM solution to this problem [34] involves the variable splitting

$$\boldsymbol{z} = \begin{pmatrix} \boldsymbol{z}_0 \\ \boldsymbol{z}_1 \end{pmatrix} = \begin{pmatrix} G_0 \\ G_1 \end{pmatrix} \boldsymbol{x} = \boldsymbol{G}\boldsymbol{x} \;, \tag{120}$$

resulting in the ADMM problem

$$\arg\min_{\boldsymbol{x},\boldsymbol{z}} \frac{1}{2}\left\|\boldsymbol{x} - \boldsymbol{d}\right\|_2^2 + w\left\|\sqrt{\boldsymbol{z}_0^2 + \boldsymbol{z}_1^2}\right\|_1$$
$$\text{s.t.} \quad \boldsymbol{z} = \boldsymbol{G}\boldsymbol{x} \;, \tag{121}$$

which corresponds to

$$\boldsymbol{A} = \boldsymbol{G} \qquad \boldsymbol{B} = -\boldsymbol{I} \qquad \boldsymbol{c} = \boldsymbol{0} \tag{122}$$

in (1).

The ADMM iterations are

$$\boldsymbol{x}^{(k+1)} = (\rho\boldsymbol{G}^T\boldsymbol{G} + \boldsymbol{I})^{-1}\left(\boldsymbol{d} + \rho\boldsymbol{G}^T\left(\boldsymbol{z}^{(k)} - \frac{\boldsymbol{y}^{(k)}}{\rho}\right)\right) \tag{123}$$

$$\boldsymbol{z}^{(k+1)} = \mathcal{R}_{w/\rho}\left(\boldsymbol{G}\boldsymbol{x}^{(k+1)} + \frac{\boldsymbol{y}^{(k)}}{\rho}\right) \tag{124}$$

$$\boldsymbol{y}^{(k+1)} = \boldsymbol{y}^{(k)} + \rho\big(\boldsymbol{G}\boldsymbol{x}^{(k+1)} - \boldsymbol{z}^{(k+1)}\big) \;, \tag{125}$$

where $\mathcal{R}$ is the block soft-thresholding operator [35, §6.5.1], applied as in [36].

Our test problem consisted of application of TV denoising to a Siemens star phantom (generated using the `xdesign` package [37]) with Gaussian white noise. Performance was again quantified in terms of the relative residual $J(\boldsymbol{x}^{(50)} - J^*)/J^*$. Results are shown in Fig. 5c and discussed in Section VI-H.

F. ℓ_1-TV Denoising

The ℓ_1 total variation (TV) denoising problem can be expressed as

$$\operatorname*{arg\,min}_{\boldsymbol{x}} \underbrace{\frac{1}{2}\left\|\boldsymbol{x}-\boldsymbol{d}\right\|_1 + w\left\|\sqrt{\left(\boldsymbol{G}_0\boldsymbol{x}\right)^2 + \left(\boldsymbol{G}_1\boldsymbol{x}\right)^2}\right\|_1}_{J(\boldsymbol{x})} , \tag{126}$$

where $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$ are defined as above. The ADMM solution to this problem [38, §2.4.4] involves the variable splitting

$$\boldsymbol{z} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{c} \, , \tag{127}$$

where

$$\boldsymbol{z} = \begin{pmatrix} \boldsymbol{z}_0 \\ \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{pmatrix} \quad \boldsymbol{G} = \begin{pmatrix} G_0 \\ G_1 \\ I \end{pmatrix} \quad \boldsymbol{c} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ -\boldsymbol{d} \end{pmatrix} \, , \tag{128}$$

resulting in the ADMM problem

$$\begin{aligned} \operatorname*{arg\,min}_{\boldsymbol{x},\boldsymbol{z}} \, & \frac{1}{2}\left\|\boldsymbol{z}_2\right\|_1 + w\left\|\sqrt{\boldsymbol{z}_0^2 + \boldsymbol{z}_1^2}\right\|_1 \\ \text{s.t.} \quad & \boldsymbol{z} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{c} \, , \end{aligned} \tag{129}$$

which corresponds to

$$\boldsymbol{A} = \boldsymbol{G} \qquad \boldsymbol{B} = -\boldsymbol{I} \tag{130}$$

in (1) (with $\boldsymbol{c}$ taking the same role here as in (1)).

The ADMM iterations are

$$\boldsymbol{x}^{(k+1)} = (\boldsymbol{G}^T\boldsymbol{G})^+\boldsymbol{G}^T\left(\boldsymbol{z}^{(k)} + \boldsymbol{c} - \frac{\boldsymbol{y}^{(k)}}{\rho}\right) \tag{131}$$

$$\begin{pmatrix} \boldsymbol{z}_0 \\ \boldsymbol{z}_1 \end{pmatrix}^{(k+1)} = \mathcal{R}_{w/\rho}\left(\begin{pmatrix} \boldsymbol{G}_0 \\ \boldsymbol{G}_1 \end{pmatrix}\boldsymbol{x}^{(k+1)} + \frac{\boldsymbol{y}^{(k)}}{\rho}\right) \tag{132}$$

$$\boldsymbol{z}_2^{(k+1)} = \mathcal{S}_{1/\rho}\left(\boldsymbol{x}^{(k+1)} - \boldsymbol{d} + \frac{\boldsymbol{y}^{(k)}}{\rho}\right) \tag{133}$$

$$\boldsymbol{y}^{(k+1)} = \boldsymbol{y}^{(k)} + \rho\big(\boldsymbol{G}\boldsymbol{x}^{(k+1)} - \boldsymbol{z}^{(k+1)} - \boldsymbol{c}\big) \, , \tag{134}$$

where $\cdot^+$ denotes the matrix pseudoinverse.

Our test problem consisted of application of ℓ_1-TV denoising to a Siemens star phantom (generated using the `xdesign` package [37]) with impulse noise. Performance was again quantified in terms of the relative residual $J(\boldsymbol{x}^{(50)} - J^*)/J^*$. Results are shown in Fig. 5d and discussed in Section VI-H.

G. Run Times

We compared the run time of the proposed method to the fixed method (i.e., standard ADMM). Because the proposed method does not involve expensive computations, we did not expect to see a large impact on run time. The results (Table 2) confirm this, with the average (taken over ρ_0) run time for the proposed method always within 10% of that of standard ADMM.

H. Summary

Our results are consistent with prior empirical observations that penalty parameter selection has a large impact on convergence. They also show that the performance of selection methods varies between optimization problems. For some problems (basis pursuit denoising, Fig. 5a), all the adaptive methods performed well; on others (robust PCA, Fig. 5b, scaled or translated quadratics, Fig. 4, and ℓ_1 denoising, Fig. 5d) there was more than a 10 times difference in relative residual between the best-performing methods and the worst. The proposed method provided consistently good performance across all experiments. The SRB method also performed well, except in the translated quadratics problem, in which it was the worst performer, which is consistent with our theoretical analysis of translation invariance. The BBS and residual balancing methods usually improved performance over using a fixed parameter, but both methods had problems where they did not offer much benefit (robust PCA and ℓ_1-TV denoising for BBS; scaled quadratics for residual balancing).

To provide another perspective on these results, we collected the performance of the methods with $\rho^{(0)} = 1.0$ in Table 3. These numbers represent performance in the scenario where the user wants to devote no effort to tuning ρ. These results show that, while no method provides the lowest relative residual across all problems, the proposed method is often the best (5 of 7 problems) and is always within an order of magnitude of the best. Every other method performs poorly for at least one problem, not providing a relative residual within one order of magnitude of the best other method.

As a different way of measuring robustness, we computed the median residual across a wide range of $\rho^{(0)}$ values (10^{-3} to 10^3) in Table 4. This approach simulates typical performance when solving a range of problems, each with a potentially different optimal $\rho^{(0)}$. The proposed method is again the best in 5 of 7 problems and is always within an order of magnitude of the best.

VII. Conclusions

In this work, we developed a new method for ADMM parameter selection. This method is based on a theoretical framework that analyses the convergence of ADMM, when applied to a quadratic problem, as an affine fixed point algorithm. While elements of this model are present in prior works (e.g. [17]), we took a fundamentally new approach to exploiting it for ADMM parameter selection by approximating the spectral radius of the iteration matrix for extreme values of the penalty parameter, rather than attempting to estimate or bound its complex behavior across the full range of penalty parameters. Based on this framework, we derived a new adaptive penalty parameter selection algorithm that we refer to as the spectral radius approximation (SRA) method. While our mathematical framework was developed for quadratic problems, the resulting algorithm does not

TABLE 2: Run time, mean ± standard deviation [s]

	method				
problem	fixed	RB	BBS	SRB	proposed
quadratics	2.0e-2 ± 2.9e-3	2.2e-2 ± 3.4e-3	2.2e-2 ± 2.8e-3	2.3e-2 ± 7.1e-3	2.2e-2 ± 4.2e-3
scaled	1.9e-2 ± 2.7e-3	2.0e-2 ± 2.2e-3	2.2e-2 ± 2.6e-3	2.1e-2 ± 2.8e-3	2.0e-2 ± 1.9e-3
translated	1.9e-2 ± 3.1e-3	2.1e-2 ± 2.3e-3	2.2e-2 ± 3.6e-3	2.0e-2 ± 2.2e-3	2.0e-2 ± 2.8e-3
robust PCA	3.4e+1 ± 1.4e+0	3.5e+1 ± 1.7e+0	3.8e+1 ± 1.8e+0	3.4e+1 ± 1.9e+0	3.5e+1 ± 2.1e+0
BPDN	1.4e-1 ± 8.2e-2	1.3e-1 ± 8.2e-3	1.1e-1 ± 2.4e-2	1.2e-1 ± 8.7e-3	1.2e-1 ± 9.9e-3
TV denoising	1.0e+0 ± 3.7e-1	1.1e+0 ± 4.4e-2	1.4e+0 ± 7.0e-2	1.1e+0 ± 4.9e-2	1.0e+0 ± 6.2e-2
ℓ_1-TV denoising	1.0e+0 ± 8.3e-2	1.1e+0 ± 4.3e-2	1.3e+0 ± 1.0e-1	1.0e+0 ± 4.6e-2	1.0e+0 ± 7.3e-2

TABLE 3: Relative residual at $k = 50$ with $\rho^{(0)} = 1.0$

	method				
problem	fixed	RB	BBS	SRB	proposed
quadratics	5.13e-4	1.96e-7	2.40e-5	8.61e-9	**1.24e-9**
scaled	4.11e-1	2.84e-1	3.96e-7	1.62e-6	**7.56e-9**
translated	3.13e-3	4.00e-6	3.26e-5	3.47e-1	**2.36e-7**
BPDN	1.35e-7	**6.73e-8**	1.35e-7	1.35e-7	1.35e-7
robust PCA	1.68e-2	1.79e-3	2.12e-3	7.36e-5	**4.76e-5**
TV denoising	7.22e-3	1.58e-3	3.75e-3	**3.51e-4**	5.03e-4
ℓ_1-TV denoising	5.39e-4	5.39e-4	7.61e-4	6.93e-4	**5.04e-4**

TABLE 4: Median relative residual at $k = 50$

	method				
problem	fixed	RB	BBS	SRB	proposed
quadratics	1.60e-1	2.36e-7	1.46e-5	3.66e-8	**3.96e-9**
scaled	4.11e-1	2.82e-1	8.73e-7	1.62e-6	**2.17e-8**
translated	2.35e-1	4.24e-7	5.83e-5	3.47e-1	**2.37e-7**
BPDN	1.58e-2	1.35e-7	**6.73e-8**	1.35e-7	**6.73e-8**
robust PCA	1.68e-2	1.39e-3	2.80e-3	7.39e-5	**4.76e-5**
TV denoising	2.02e-2	2.05e-3	2.57e-3	**3.51e-4**	4.48e-4
ℓ_1-TV denoising	2.42e-1	5.14e-4	7.40e-1	7.94e-4	**4.31e-4**

make explicit use of the quadratic structure and can therefore be applied to any ADMM problem, which we view as a making an implicit iterative local quadratic approximation. The SRA method is simple to implement and enjoys theoretical advantages over all prior methods: it is scaling covariant, while residual balancing [12] is not; it is translation invariant, while the spectral radius bound method of [17] is not; and it uses a more general model of the optimization problem than the Barzilai-Borwein spectral method of [15]. This framework also allowed us to present new interpretations of prior methods that provide useful insights into their relative advantages and disadvantages. Finally, our proposed method exhibits empirical performance that is competitive with—and often superior to—state-of-the-art comparison methods.

Appendix A
Scaled Form ADMM

It is often convenient to write ADMM in a scaled form by making the substitution $\boldsymbol{u} = \rho^{-1}\boldsymbol{y}$ [5, §3.1.1]. Adaptive ADMM in the scaled form can be written as

$$\boldsymbol{x}^{(k+1)} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{\rho^{(k)}}{2} \left\| \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z}^{(k)} - \boldsymbol{c} + \boldsymbol{u}^{(k)} \right\|^2 \tag{135}$$

$$\boldsymbol{z}^{(k+1)} = \arg\min_{\boldsymbol{z}} g(\boldsymbol{z}) + \frac{\rho^{(k)}}{2} \left\| \boldsymbol{A}\boldsymbol{x}^{(k+1)} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{c} + \boldsymbol{u}^{(k)} \right\|^2 \tag{136}$$

$$\rho^{(k+1)} = \phi\Big(\big(\rho^{(j)}, \boldsymbol{x}^{(j+1)}, \boldsymbol{z}^{(j+1)}, \boldsymbol{u}^{(j)}\big)_{j=0}^{k}\Big) \tag{137}$$

$$\boldsymbol{u}^{(k+1)} = \frac{\rho^{(k)}}{\rho^{(k+1)}} \left(\boldsymbol{u}^{(k)} + \left(\boldsymbol{A}\boldsymbol{x}^{(k+1)} + \boldsymbol{B}\boldsymbol{z}^{(k+1)} - \boldsymbol{c}\right)\right) , \tag{138}$$

where we have made the rescaling of the dual variable when ρ changes explicit in (138). The proposed method may be applied in this scaled form by using Algorithm 2 for the function ϕ in (137).

Algorithm 2: Proposed ρ selection method (scaled dual variable version)

```
Input: $k, \rho^{(k)}, \boldsymbol{x}^{(k+1)}, \boldsymbol{z}^{(k)}, \boldsymbol{z}^{(k+1)}$
Parameters: $T = 5, \tau^{\text{incr}} = \tau^{\text{decr}} = 10$
Output: $\rho^{(k+1)}$
if $k \mod T \neq 1$ :
    return $\rho^{(k)}$
$p \leftarrow \left\|\rho^{(k)}(\boldsymbol{A}\boldsymbol{x}^{(k+1)} + \boldsymbol{B}\boldsymbol{z}^{(k+1)} - \boldsymbol{c})\right\|$
$q \leftarrow \left\|\boldsymbol{B}\big(\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)}\big)\right\|$
if $p = 0$ and $q > 0$ :
    return $\rho^{(k)}/\tau^{\text{decr}}$
if $p > 0$ and $q = 0$ :
    return $\tau^{\text{incr}}\rho^{(k)}$
if $p = 0$ and $q = 0$ :
    return $\rho^{(k)}$
return $p/q$
```

Appendix B
Equivalence of Section II-A to DRS on the dual

The equivalent version of our F (11) and G (8) follow from (11) in [15]

$$\begin{aligned}\boldsymbol{A}\boldsymbol{x}^{(k+1)} - \boldsymbol{c} \in \partial\hat{H}(-\tilde{\boldsymbol{y}}^{(k+1)}) \Rightarrow \\ \partial\hat{H}\big(-\tilde{\boldsymbol{y}}^{(k+1)}\big) = \boldsymbol{A}F\big(\tilde{\boldsymbol{y}}^{(k+1)}\big) - \boldsymbol{c}\end{aligned} \tag{139}$$

and

$$\begin{aligned}\boldsymbol{B}\boldsymbol{z}^{(k+1)} \in \partial\hat{G}\big(-\boldsymbol{y}^{(k+1)}\big) \Rightarrow \\ \partial\hat{G}\big(-\boldsymbol{y}^{(k+1)}\big) = \boldsymbol{B}G\big(\boldsymbol{y}^{(k+1)}\big)\ .\end{aligned} \tag{140}$$

Translating the first DRS step ((12) in [15]) gives

$$\begin{aligned}-\tilde{\boldsymbol{y}}^{(k+1)} &= -\boldsymbol{y}^{(k)} - \rho\big(\boldsymbol{A}F\big(\tilde{\boldsymbol{y}}^{(k+1)}\big) - \boldsymbol{c} + \boldsymbol{B}G\big(\boldsymbol{y}^{(k)}\big)\big)\\ \tilde{\boldsymbol{y}}^{(k+1)} &= \boldsymbol{y}^{(k)} + \rho\big(\boldsymbol{A}F\big(\tilde{\boldsymbol{y}}^{(k+1)}\big) - \boldsymbol{c} + \boldsymbol{B}G\big(\boldsymbol{y}^{(k)}\big)\big)\\ \tilde{\boldsymbol{y}}^{(k+1)} - \rho\boldsymbol{A}F\big(\tilde{\boldsymbol{y}}^{(k+1)}\big) &= \boldsymbol{y}^{(k)} + \rho\boldsymbol{B}G\big(\boldsymbol{y}^{(k)}\big) - \rho\boldsymbol{c}\\ \tilde{\boldsymbol{y}}^{(k+1)} &= (\boldsymbol{I} - \rho\boldsymbol{A}F)^{+}\big(\boldsymbol{y}^{(k)} + \rho\boldsymbol{B}G(\boldsymbol{y}^{(k)}) - \rho\boldsymbol{c}\big)\ ,\end{aligned} \tag{141}$$

which matches the first line of (18). Translating the second DRS step ((13) in [15]) gives

$$\begin{aligned}-\boldsymbol{y}^{(k+1)} &= -\boldsymbol{y}^{(k)} - \rho\big(\boldsymbol{A}F\big(\tilde{\boldsymbol{y}}^{(k+1)}\big) - \boldsymbol{c} + \boldsymbol{B}G\big(\boldsymbol{y}^{(k+1)}\big)\big)\\ \boldsymbol{y}^{(k+1)} &= \boldsymbol{y}^{(k)} + \rho\big(\boldsymbol{A}F\big(\tilde{\boldsymbol{y}}^{(k+1)}\big) - \boldsymbol{c} + \boldsymbol{B}G\big(\boldsymbol{y}^{(k+1)}\big)\big)\\ \boldsymbol{y}^{(k+1)} - \rho\boldsymbol{B}G\big(\boldsymbol{y}^{(k+1)}\big) &= \boldsymbol{y}^{(k)} + \rho\boldsymbol{A}F\big(\tilde{\boldsymbol{y}}^{(k+1)}\big) - \rho\boldsymbol{c}\\ \boldsymbol{y}^{(k+1)} &= (\boldsymbol{I} - \rho\boldsymbol{B}G)^{+}\big(\boldsymbol{y}^{(k)} + \rho\boldsymbol{A}F\big(\tilde{\boldsymbol{y}}^{(k+1)}\big) - \rho\boldsymbol{c}\big)\ .\end{aligned} \tag{142}$$

We then use the derivation involving $\tilde{\boldsymbol{y}}$ to express $\rho\boldsymbol{A}F\big(\tilde{\boldsymbol{y}}\big)$ in terms of $\tilde{\boldsymbol{y}}$ and $\boldsymbol{y}$:

$$\rho\boldsymbol{A}F\big(\tilde{\boldsymbol{y}}^{(k+1)}\big) = \tilde{\boldsymbol{y}}^{(k+1)} - \boldsymbol{y}^{(k)} - \rho\boldsymbol{B}G\big(\boldsymbol{y}^{(k)}\big) + \rho\boldsymbol{c} \tag{143}$$

and therefore

$$\boldsymbol{y}^{(k+1)} = (\boldsymbol{I} - \rho\boldsymbol{B}G)^{+}\big(\tilde{\boldsymbol{y}}^{(k+1)} - \rho\boldsymbol{B}G\big(\boldsymbol{y}^{(k)}\big)\big)\ , \tag{144}$$

which matches the second line of (18). Thus we have shown that the expression of ADMM as DRS on the dual in [15] results in the same iteration on $\boldsymbol{y}$ that we derived by working with optimality conditions of the ADMM steps.

REFERENCES

[1] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Springer Optimization and Its Applications*. Springer New York, 2011, pp. 185–212. doi:10.1007/978-1-4419-9569-8_10

[2] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires," *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, vol. 9, no. R2, pp. 41–76, Aug. 1975.

[3] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976. doi:10.1016/0898-1221(76)90003-1

[4] L. Li, X. Wang, and G. Wang, "Alternating direction method of multipliers for separable convex optimization of real functions in complex variables," *Mathematical Problems in Engineering*, vol. 2015, Dec. 2015. doi:10.1155/2015/104531

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jan. 2011. doi:10.1561/2200000016

[6] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *Journal of Scientific Computing*, vol. 66, no. 3, pp. 889–916, May 2015. doi:10.1007/s10915-015-0048-x

[7] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, pp. 29–63, 2019. doi:10.1007/s10915-018-0757-z

[8] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 644–658, Mar. 2015. doi:10.1109/TAC.2014.2354892

[9] A. U. Raghunathan and S. Di Cairano, "Alternating direction method of multipliers for strictly convex quadratic programs: optimal parameter selection," in *American Control Conference (ACC)*, Jun. 2014, pp. 4324–4329. doi:10.1109/ACC.2014.6859093

[10] ——, "ADMM for convex quadratic programs: Linear convergence and infeasibility detection," Oct. 2015," `arXiv:1411.7288`.

[11] Y. Lin, B. Wohlberg, and V. Vesselinov, "ADMM penalty parameter selection with Krylov subspace recycling technique for sparse coding," in *IEEE International Conference on Image Processing (ICIP)*, Beijing, China, Sep. 2017, pp. 1945–1949. doi:10.1109/ICIP.2017.8296621

[12] B.-S. He, H. Yang, and S.-L. Wang, "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities," *Journal of Optimization Theory and Applications*, vol. 106, pp. 337–356, 2000. doi:10.1023/a:1004603514434

[13] B. Wohlberg, "ADMM penalty parameter selection by residual balancing," Apr. 2017," `arXiv:1704.06209`.

[14] M. Yan and W. Yin, "Self equivalence of the alternating direction method of multipliers," *Splitting Methods in Communication, Imaging, Science, and Engineering*, pp. 165–194, 2016. doi:10.1007/978-3-319-41589-5_5

[15] Z. Xu, M. Figueiredo, and T. Goldstein, "Adaptive ADMM with Spectral Penalty Parameter Selection," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 54, Apr. 2017, pp. 718–727.

[16] Z. Xu, M. A. T. Figueiredo, X. Yuan, C. Studer, and T. Goldstein, "Adaptive relaxed ADMM: Convergence theory and practical implementation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017. doi:10.1109/cvpr.2017.765

[17] D. A. Lorenz and Q. Tran-Dinh, "Non-stationary Douglas-Rachford and alternating direction method of multipliers: adaptive step-sizes and convergence," *Computational Optimization and Applications*, vol. 74, no. 1, pp. 67–92, May 2019. doi:10.1007/s10589-019-00106-9

[18] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. Jordan, "A general analysis of the convergence of ADMM," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 37, Lille, France, 07–09 Jul 2015, pp. 343–352.

[19] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 57–95, Jan. 2016. doi:10.1137/15m1009597

[20] Y. Nesterov, *Introductory Lectures on Convex Optimization*. Springer US, 2004.

[21] B. Bollobás, *Linear Analysis: An Introductory Course*, 2nd ed. Cambridge University Press, 1999.

[22] G. H. Golub and C. F. V. Loan, *Matrix Computations*. JHU Press, 2013.

[23] B.-S. He, H. Yang, and S.-L. Wang, "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities," *Journal of Optimization Theory and Applications*, vol. 106, pp. 337–356, 2000. doi:10.1023/a:1004603514434

[24] A. Hansson, Z. Liu, and L. Vandenberghe, "Subspace system identification via weighted nuclear norm optimization," in *IEEE Conference on Decision and Control (CDC)*, Dec. 2012. doi:10.1109/CDC.2012.6426980

[25] Z. Liu, A. Hansson, and L. Vandenberghe, "Nuclear norm system identification with missing inputs and outputs," *Systems & Control Letters*, vol. 62, no. 8, pp. 605 – 612, 2013. doi:10.1016/j.sysconle.2013.04.005

[26] V. Q. Vu, J. Cho, J. Lei, and K. Rohe, "Fantope projection and selection: A near-optimal convex relaxation of sparse PCA," in *Advances in Neural Information Processing Systems 26*, 2013, pp. 2670–2678.

[27] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Collaborative sparse regression for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 341–354, Jan. 2014. doi:10.1109/TGRS.2013.2240001

[28] D. S. Weller, A. Pnueli, O. Radzyner, G. Divon, Y. C. Eldar, and J. A. Fessler, "Phase retrieval of sparse signals using optimization transfer and ADMM," in *IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 1342–1346.

[29] B. Wohlberg, "Efficient convolutional sparse coding," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 7173–7177. doi:10.1109/ICASSP.2014.6854992

[30] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001. doi:10.1137/S003614450037906X

[31] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, Apr. 2004. doi:10.1214/009053604000000067

[32] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, pp. 1–37, 2011. doi:10.1145/1970392.1970395

[33] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010. doi:10.1137/080738970

[34] T. Goldstein and S. J. Osher, "The split Bregman method for L1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009. doi:10.1137/080725891

[35] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014. doi:10.1561/2400000003

[36] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM Journal on Imaging Sciences*, vol. 1, no. 3, pp. 248–272, 2008. doi:10.1137/080724265

[37] D. J. Ching and D. Gürsoy, "XDesign: an open-source software package for designing X-ray imaging phantoms and experiments," *Journal of Synchrotron Radiation*, vol. 24, no. 2, pp. 537–544, Mar. 2017. doi:10.1107/S1600577517001928

[38] E. Esser, "Primal dual algorithms for convex models and applications to image restoration, registration and nonlocal inpainting," Ph.D. dissertation, University of California Los Angeles, 2010.