FAST ALTERNATING DIRECTION OPTIMIZATION METHODS

TOM GOLDSTEIN, BRENDAN O'DONOGHUE, SIMON SETZER, AND RICHARD BARANIUK

Abstract. Alternating direction methods are a common tool for general mathematical programming and optimization. These methods have become particularly important in the field of variational image processing, which frequently requires the minimization of non-differentiable objectives. This paper considers accelerated (i.e., fast) variants of two common alternating direction methods: the Alternating Direction Method of Multipliers (ADMM) and the Alternating Minimization Algorithm (AMA). The proposed acceleration is of the form first proposed by Nesterov for gradient descent methods. In the case that the objective function is strongly convex, global convergence bounds are provided for both classical and accelerated variants of the methods. Numerical examples are presented to demonstrate the superior performance of the fast methods for a wide variety of problems.

Contents

1. Introduction

This manuscript considers the problem

$$\begin{array}{ll} \text{minimize} & H(u)+G(v) \\ \text{subject to} & Au+Bv=b \end{array} \tag{1}$$

over variables $u \in R^{N_u}$ and $v \in R^{N_v}$ where $H : R^{N_u} \to (-\infty, \infty]$ and $G : R^{N_v} \to (-\infty, \infty]$ are closed convex functions, $A \in R^{N_b \times N_u}$ and $B \in R^{N_b \times N_v}$ are linear operators, and $b \in R^{N_b}$ is a vector of data. In many cases H and G each have a simple, exploitable structure. In this case alternating direction methods, whereby we minimize over one function and then the other, are efficient simple to implement.

The formulation (1) arises naturally in many application areas. One common application is ℓ_1 regularization for the solution of linear inverse problems. In statistics, ℓ_1-penalized least-squares problems are often used for "variable selection," which enforces that only a small subset of the regression parameters are non-zero [1, 2, 3, 4, 5, 6, 7]. In the context of image processing, one commonly solves variational models involving the total variation semi-norm [8]. In this case, the regularizer is the ℓ_1 norm of the gradient of an image.

Two common splitting methods for solving problem (1) are the Alternating Direction Method of Multipliers (ADMM) and the Alternating Minimization Algorithm (AMA). Both of the these schemes solve (1) using a sequence of steps that decouple H and G. While ADMM and AMA result in simple algorithms for solving (1), they often perform poorly in situations where the components of (1) are poorly conditioned or when high precision is required.

This paper presents new accelerated variants of ADMM and AMA that exhibit faster convergence than conventional splitting methods. Convergence rates are given for the accelerated methods in the case that the problems are strongly convex. For general problems of the form (1) (which may not be strongly convex), we present "restart" rules that allow the acceleration to be applied with guaranteed convergence.

1.1. Structure of this Paper.

In Section 2 we present the ADMM and AMA splitting methods, and provide a review of currently available accelerated optimization schemes. In Section 3 we develop a global convergence theory for non-accelerated ADMM. In Section 4, we present the new accelerated ADMM scheme. Global convergence bounds are provided for the case when the objective is strongly convex, and a rate is given. In the case of weakly convex functions we present a "restart" scheme that is globally convergent but without a guaranteed rate. In Section 5 we present an accelerated variant of AMA. We discuss the relationship of the proposed methods to previous schemes in Section 6. Finally, we present numerical results demonstrating the effectiveness of the proposed acceleration for image restoration, compressed sensing, and quadratic programming problems in Section 7.

1.2. Terminology and Notation.

Given some tuple $x = \{x_i | i \in \Omega\}$ with index set Ω. We denote the ℓ_2-norm and ℓ_1-norm as follows:

$$\|x\| = \sqrt{\sum_{i=1}^{n} |x_i|^2}, \qquad |x| = \sum_{i=1}^{n} |x_i|,$$

respectively.

In much of the theory below, we shall make use of the *resolvent* operator, as described by Moreau [9]. Let $\partial F(x)$ be the subdifferential of a convex function F at x. The resolvent (or proximal mapping) of a maximal monotone operator F is:

$$J_{\partial F}(z) := (I + \partial F)^{-1} z = \operatorname*{argmin}_x F(x) + \frac{1}{2}\|x - z\|^2.$$

For example, when $F(x) = \mu|x|$ for some $\mu > 0$ we have

$$J_{\partial F}(z) := \mathbf{shrink}(z, \mu)$$

where the i^{th} entry of the 'shrink' function is

$$\mathbf{shrink}(z,\mu)_i := \frac{z_i}{|z_i|}\max\{|z_i| - \mu, 0\} = \max\{|z_i| - \mu, 0\}\mathbf{sign}\{z_i\}. \tag{2}$$

We shall also make extensive use of the *conjugate* of a convex function. The conjugate of a convex function F, denoted F^*, is defined as

$$F^*(p) = \sup_u \langle u, p \rangle - F(u).$$

The conjugate function satisfies the important identity

$$p \in \partial F(u) \Leftrightarrow u \in \partial F^*(p)$$

that will be used extensively below. For an in-depth discussion of the conjugate function and its properties, see [10, 11].

We shall often refer to *strongly convex* functions. If H is strongly convex then there exists a constant $\sigma_H > 0$ such that for every $x, y \in R^{N_u}$

$$\langle p - q, x - y \rangle \geq \sigma_H \|x - y\|^2,$$

where $p \in \partial H(x)$ and $q \in \partial H(y)$. This can be written equivalently as

$$\lambda H(x) + (1 - \lambda) H(y) - H(\lambda x + (1 - \lambda) y) \geq \sigma_H \lambda (1 - \lambda) \|x - y\|^2$$

for $\lambda \in [0, 1]$. Intuitively, strong convexity means that a function lies above its local quadratic approximation.

If H is strongly convex, then its conjugate function has a Lipschitz continuous gradient, in which case $\partial H^*(y) = \{\nabla H^*(y)\}$. If H is strongly convex with modulus σ_H, then

$$L(\nabla H^*) = \sigma_H^{-1},$$

where $L(\nabla H^*)$ denotes the Lipschitz constant of ∇H, *i.e.*,

$$\|\nabla H^*(x) - \nabla H^*(y)\| \leq L(\nabla H^*) \|x - y\|.$$

For an extensive discussion of the properties of strongly convex functions see Proposition 12.60 of [11].

We will work frequently with the dual of problem (1), which we write as

$$\text{maximize} \quad \Big(D(\lambda) = -H^*(A^T \lambda) + \langle \lambda, b \rangle - G^*(B^T \lambda) \Big) \tag{3}$$

over $\lambda \in R^{N_b}$ and where $D : R^{N_b} \to R$ is the concave dual objective. We denote by $\lambda^\star$ the optimum of the above problem (presuming it exists) and by $D(\lambda^\star)$ the optimal objective value.

Finally, we denote the spectral radius of a matrix A by $\rho(A)$. Using this notation the matrix norm of A can be written $\sqrt{\rho(A^T A)}$.

1.3. **Alternating Direction Methods.** Alternating direction methods enable problems of the form (1) to be solved by addressing H and G separately.

The *Alternating Direction Method of Multipliers* (ADMM) solves the coupled problem (1) using the uncoupled steps listed in Algorithm 1:

Algorithm 1 ADMM

```
Require: $v_0 \in R^{N_v}$, $\lambda_0 \in R^{N}_{b}$, $\tau > 0$
for $k = 0, 1, 2, \ldots$ do
    $u_{k+1} = \text{argmin}_u \, H(u) + \langle \lambda_k, -Au \rangle + \frac{\tau}{2} \|b - Au - Bv_k\|^2$
    $v_{k+1} = \text{argmin}_v \, G(v) + \langle \lambda_k, -Bv \rangle + \frac{\tau}{2} \|b - Au_{k+1} - Bv\|^2$
    $\lambda_{k+1} = \lambda_k + \tau(b - Au_{k+1} - Bv_{k+1})$
end for
```

The ADMM was first described by Glowinski and Marocco [12]. Convergence results were studied by Gabay [13], Glowinski and Le Tallec [14], as well as Eckstein and Bertsekas [15]. In the context of ℓ_1 regularized problems, this technique is commonly known as the *Split Bregman Method* [16], and is known to be an efficient solver for problems involving the total-variation norm [17].

Another well-known alternating direction method is Tseng's *Alternating Minimization Algorithm* (AMA) [18]. This method has simpler steps than ADMM, but requires strong convexity of one of the objectives. See Algorithm 2.

Algorithm 2 AMA

```
Require: $\lambda_0 \in R^{N_b}$, $\tau > 0$
for $k = 0, 1, 2, \ldots$ do
    $u_{k+1} = \text{argmin}_u H(u) + \langle \lambda_k, -Au \rangle$
    $v_{k+1} = \text{argmin}_v G(v) + \langle \lambda_k, -Bv \rangle + \frac{\tau}{2}\|b - Au_{k+1} - Bv\|^2$
    $\lambda_{k+1} = \lambda_k + \tau(b - Au_{k+1} - Bv_{k+1})$
end for
```

The goal of this paper is to develop accelerated (i.e., fast) variants of the ADMM and AMA methods. We will show that the ADMM and AMA iterations can be accelerated using techniques of the type first proposed by Nesterov [19]. Nesterov's method was originally intended to accelerate the minimization of convex functions using first order (e.g., gradient descent type) methods. In the context of alternating direction methods, we will show how the technique can be used to solve the constrained problem (1).

1.4. **Optimality Conditions and Residuals for ADMM.** The solution to a wide class of problems can be described by the KKT conditions [10]. In the case of the constrained problem (1), there are two conditions that describe an optimal solution (see [20] and [10] for an in-depth treatment of optimality conditions for this problem). First, the primal variables must be feasible. This results in the condition

$$b - Au^\star - Bv^\star = 0. \tag{4}$$

Next, the dual variables must satisfy the Lagrange multiplier (or dual feasibility) condition

$$0 \in \partial H(u^\star) - A^T\lambda^\star, \tag{5}$$

$$0 \in \partial G(v^\star) - B^T\lambda^\star. \tag{6}$$

In general, it could be that G or H are non-differentiable, in which case these optimality conditions are difficult to check. Using the optimality conditions for the individual steps of Algorithm 1, we can arrive at more useful expressions for these quantities. From the optimality condition for Step 3 of Algorithm 1, we have

$$\begin{aligned} 0 \in\ & \partial G(v_{k+1}) - B^T\lambda_k - \tau B^T(b - Au_{k+1} - Bv_{k+1}) \\ & = \partial G(v_{k+1}) - B^T\lambda_{k+1}. \end{aligned}$$

Thus, the second dual optimality condition (5) is satisfied by v_{k+1} and λ_{k+1} at the end of each iteration of Algorithm 1.

To obtain a useful expression for the condition (5), we use the optimality condition for Step 2 of Algorithm 1,

$$0 \in \partial H(u_{k+1}) - A^T\lambda_k - \tau A^T(b - Au_{k+1} - Bv_k) = \partial H(u_{k+1}) - A^T\lambda_{k+1} - \tau A^T B(v_{k+1} - v_k).$$

After each iteration of Algorithm 1, we then have

$$\tau A^T B(v_{k+1} - v_k) \in \partial H(u_{k+1}) - A^T\lambda_{k+1}.$$

One common way to measure how well the iterates of Algorithm 1 satisfy the KKT conditions is to define the primal and dual residuals:

$$r_k = b - Au_k - Bv_k, \tag{7}$$

$$d_k = \tau A^T B(v_k - v_{k-1}). \tag{8}$$

The size of these residuals indicates how far the iterates are from a solution. One of the goals of this article is to establish accelerated versions of Algorithm (1) such that these residuals decay quickly.

1.5. **Optimality Conditions and Residuals for AMA.** We can write the KKT condition (5) in terms of the iterates of AMA using the optimality condition for Step 2 of Algorithm 2, which is

$$0 \in \partial H(u_{k+1}) - A^T\lambda_k = \big(\partial H(u_{k+1}) - A^T\lambda_{k+1}\big) + A^T(\lambda_{k+1} - \lambda_k).$$

The residuals for AMA can thus be written

$$r_k = b - Au_k - Bv_k, \tag{9}$$

$$d_k = A^T(\lambda_k - \lambda_{k+1}). \tag{10}$$

2. Optimal Descent Methods

Consider the unconstrained problem

$$\text{minimize} \quad F(x) \tag{11}$$

over variable $x \in R^N$, for some Lipschitz continuously differentiable, convex function $F : R^N \to R$. We consider standard first-order methods for solving this type of problem. We begin with the method of gradient descent described in Algorithm 3.

Algorithm 3 Gradient Descent

```
for $k = 0, 1, 2, \cdots$ do
    $x_{k+1} = x_k - \tau\nabla F(x_k)$
end for
```

Convergence of gradient descent is guaranteed as long as $\tau < \frac{2}{L(\nabla F)}$, where $L(\nabla F)$ is the Lipschitz constant for ∇F.

In case $F = H + G$ is the sum of two convex functions, a very useful minimization technique is the forward-backward splitting (FBS) method (Algorithm 4), first introduced by Bruck [21] and popularized by Passty [22].

Algorithm 4 FBS

```
for $k = 1, 2, 3, \ldots$ do
    $x_{k+1} = J_{\tau\partial G}(x_k - \tau\partial H(x_k))$
end for
```

FBS has fairly general convergence properties. In particular, it is convergent as long as $\tau < 2/L(\partial H)$ [22]. See [23] for a more recent and in-depth analysis of FBS.

The complexity of first-order methods has been studied extensively by Nemirovsky and Yudin [24]. For the general class of problems with Lipschitz continuous derivative, gradient descent achieves the global convergence rate $O(1/k)$, meaning that $F(x_k) - F^* < C/k$ for some constant $C > 0$ [25]. Similar $O(1/k)$ convergence bounds have been provided for more sophisticated first-order methods including the proximal point method (PPM) [26] and FBS [27].

In a seminal paper, Nesterov [19] presented a first-order minimization scheme with $O(1/k^2)$ global convergence — a rate which is provably optimal for the class of Lipschitz differentiable functionals [25]. Nesterov's optimal method is a variant of gradient descent which is accelerated by an over-relaxation step; see Algorithm 5.

Algorithm 5 Nesterov's Accelerated Gradient Descent

```
Require: $\alpha_0 = 1$, $x_0 = y_1 \in R^N$, $\tau < 1/L(\nabla F)$
for $k = 1, 2, 3 \ldots$ do
    $x_k = y_k - \tau\nabla F(y_k)$
    $\alpha_{k+1} = (1 + \sqrt{4\alpha_k^2 + 1})/2$
    $y_{k+1} = x_k + (\alpha_k - 1)(x_k - x_{k-1})/\alpha_{k+1}$
end for
```

Since the introduction of this optimal method, much work has been done applying Nesterov's concept to other first-order splitting methods. Nesterov himself showed that $O(1/k^2)$ complexity results can be proved for certain classes of non-differentiable problems [28]. Nesterov's arguments have also been adapted to the acceleration of proximal-point descent methods by Güler [26]. Like Nesterov's gradient descent method, Güler's method achieves $O(1/k^2)$ global convergence, which is provably optimal.

More recently, there has been reinvigorated interest in optimal first-order schemes in the context of compressed sensing [29, 30]. One particular algorithm of interest is the optimal forward-backward splitting of Beck and Teboulle [27], which they have termed FISTA. See Algorithm 6.

Algorithm 6 FISTA

```
Require: $y_1 = x_0 \in R^N$, $\alpha_1 = 1$, $\tau < 1/L(\nabla G)$
for $k = 1, 2, 3, \dots$ do
    $x_k = J_{\tau \partial G}(y_k - \tau \nabla H(y_k))$
    $\alpha_{k+1} = (1 + \sqrt{1 + 4\alpha_k^2})/2$
    $y_{k+1} = x_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(x_k - x_{k-1})$
end for
```

FISTA is one the first examples of global acceleration being applied to a splitting scheme. However, the method cannot directly solve saddle point problems involving Lagrange multipliers (i.e., problems involving linear constraints).

One approach for solving saddle point problems is the accelerated alternating direction method proposed by Goldfarb et al. [31]. The authors of [31] propose a Nesterov-type acceleration for problems of the form (1) in the special case where both A and B are identity matrices, and one of H or G is differentiable. This scheme is based on the "symmetric" ADMM method, which differs from Algorithm 1 in that it involves two dual updates per iteration rather than one. The authors handle weakly convex problems by introducing a step-skipping process that applies the acceleration selectively on certain iterations. These "step-skipping" methods have a more complicated sequence of steps than conventional ADMM, but are necessary to maintain stability of the method in the presence of a Nesterov-type acceleration step.

3. Global Convergence Bounds for Unaccelerated ADMM

We now develop global convergence bounds for ADMM. We consider the case where both terms in the objective are strongly convex:

Assumption 1. *Both H and G are strongly convex with moduli σ_H and σ_G, respectively.*

Note that this assumption is rather strong – in many common problems solved using ADMM, one or both of the objective terms are not strongly convex. Later, we shall see how convergence can be guaranteed for general objectives by incorporating a simple restart rule.

As we shall demonstrate below, the conventional (unaccelerated) ADMM converges in the dual objective with rate $O(1/k)$. Later, we shall introduce an accelerated variant that converges with the optimal bound $O(1/k^2)$.

3.1. Preliminary Results.

Before we prove global convergence results for ADMM, we require some preliminary definitions and lemmas.

To simplify notation, we define the "pushforward" operators, u^+, v^+, and λ^+ for ADMM such that

$$u^+ = \operatorname*{argmin}_u H(u) + \langle \lambda, -Au \rangle + \frac{\tau}{2}\|b - Au - Bv\|^2, \tag{12}$$

$$v^+ = \operatorname*{argmin}_z G(z) + \langle \lambda, -Bz \rangle + \frac{\tau}{2}\|b - Au^+ - Bz\|^2, \tag{13}$$

$$\lambda^+ = \lambda + \tau(b - Au^+ - Bv^+). \tag{14}$$

In plain words, u^+, v^+, and λ^+ represent the results of an ADMM iteration starting with v and λ. Note that that these pushforward operators are implicitly functions of v and λ, but we have omitted this dependence for brevity.

Again we consider the dual to problem (1) which we restate here for clarity:

$$\text{maximize} \quad \Big(D(\lambda) = -H^*(A^T\lambda) + \langle \lambda, b\rangle - G^*(B^T\lambda)\Big). \tag{15}$$

Define the following operators:

$$\Psi(\lambda) \quad = \quad A\nabla H^*(A^T\lambda), \tag{16}$$
$$\Phi(\lambda) \quad = \quad B\nabla G^*(B^T\lambda). \tag{17}$$

Note that the derivative of the dual functional is $b - \Psi - \Phi$. Maximizing the dual problem is therefore equivalent to finding a solution to $b \in \Psi(\lambda^\star) + \Phi(\lambda^\star)$. Note that Assumption 1 guarantees that both Ψ and Φ are Lipschitz continuous with $L(\Psi) \le \frac{\rho(A^TA)}{\sigma_H}$ and $L(\Phi) \le \frac{\rho(B^TB)}{\sigma_G}$.

Lemma 1. *Let $\lambda \in R^{N_b}, v \in R^{N_v}$. Define*

$$\begin{aligned}\lambda^{\frac{1}{2}} &= \lambda + \tau(b - Au^+ - Bv),\\ \lambda^+ &= \lambda + \tau(b - Au^+ - Bv^+).\end{aligned}$$

Then we have

$$\begin{aligned}Au^+ &= A\nabla H^*(A^T\lambda^{\frac{1}{2}}) := \Psi(\lambda^{\frac{1}{2}}),\\ Bv^+ &= B\nabla G^*(B^T\lambda^+) := \Phi(\lambda^+).\end{aligned}$$

Proof. From the optimality condition for u^+ given in equation (12),

$$\begin{aligned}0 \;&\in\; \partial H(u^+) - A^T\lambda - \tau A^T(b - Au^+ - Bv)\\ &\;\;= \partial H(u^+) - A^T\lambda^{\frac{1}{2}}.\end{aligned}$$

From this, we obtain

$$A^T\lambda^{\frac{1}{2}} \in \partial H(u^+) \Leftrightarrow u^+ = \nabla H^*(A^T\lambda^{\frac{1}{2}}) \Rightarrow Au^+ = A\nabla H^*(A^T\lambda^{\frac{1}{2}}) = \Psi(\lambda^{\frac{1}{2}}).$$

From the optimality condition for v^+ given in equation (13),

$$\begin{aligned}0 \;&\in\; \partial G(v^+) - B^T\lambda - \tau B^T(b - Au^+ - Bv^+)\\ &\;\;= \partial G(v^+) - B^T\lambda^+,\end{aligned}$$

which yields

$$B^T\lambda^+ \in \partial G(v^+) \Leftrightarrow v^+ = \nabla G^*(B^T\lambda^+) \Rightarrow Bv^+ = B\nabla G^*(B^T\lambda^+) = \Phi(\lambda^+).$$

□

The following key lemma provides bounds on the dual functional and its value for successive iterates of the dual variable.

Lemma 2. *Suppose that $\tau^3 \le \frac{\sigma_H\sigma_G^2}{\rho(A^TA)\rho(B^TB)^2}$ and that $Bv = \Phi(\lambda)$. Then, for any $\gamma \in R^{N_b}$*

$$D(\lambda^+) - D(\gamma) \ge \tau^{-1}\langle \gamma - \lambda, \lambda - \lambda^+\rangle + \frac{1}{2\tau}\|\lambda - \lambda^+\|^2.$$

Proof. Once again, let $\lambda^{\frac{1}{2}} = \lambda + \tau(b - Au^+ - Bv)$. Define the constants $\alpha := \frac{\rho(A^TA)}{\sigma_H} \ge L(\Psi)$, and $\beta := \frac{\rho(B^TB)}{\sigma_G} \ge L(\Phi)$. Choose a value of τ such that $\tau^3 \le \frac{1}{\alpha\beta^2}$. We begin by noting that

$$\begin{aligned}\|\lambda^+ - \lambda^{\frac{1}{2}}\|^2 &= \tau^2\|Bv^+ - Bv\|^2 = \tau^2\|\Phi(\lambda^+) - \Phi(\lambda)\|^2\\ &\le \tau^2\beta^2\|\lambda^+ - \lambda\|^2.\end{aligned}$$

We now derive some tasty estimates for H and G. We have

$$\begin{aligned}
H^*(A^T\gamma) - H^*(A^T\lambda^+) &= H^*(A^T\gamma) - H^*(A^T\lambda^{\frac{1}{2}}) + H^*(A^T\lambda^{\frac{1}{2}}) - H^*(A^T\lambda^+) \qquad (18)\\
&\geq H^*(A^T\lambda^{\frac{1}{2}}) + \langle \gamma - \lambda^{\frac{1}{2}}, A\nabla H^*(A^T\lambda^{\frac{1}{2}})\rangle \\
&\quad - \Big(H^*(A^T\lambda^{\frac{1}{2}}) + \langle \lambda^+ - \lambda^{\frac{1}{2}}, A\nabla H^*(A^T\lambda^{\frac{1}{2}})\rangle + \frac{\alpha}{2}\|\lambda^+ - \lambda^{\frac{1}{2}}\|^2\Big) \\
&= \langle \gamma - \lambda^+, \Psi(\lambda^{\frac{1}{2}})\rangle - \frac{\alpha}{2}\|\lambda^+ - \lambda^{\frac{1}{2}}\|^2 \\
&\geq \langle \gamma - \lambda^+, \Psi(\lambda^{\frac{1}{2}})\rangle - \frac{\alpha\tau^2\beta^2}{2}\|\lambda^+ - \lambda\|^2 \\
&\geq \langle \gamma - \lambda^+, \Psi(\lambda^{\frac{1}{2}})\rangle - \frac{1}{2\tau}\|\lambda^+ - \lambda\|^2, \qquad (19)
\end{aligned}$$

where we have used the assumption $\tau^3 \leq \frac{1}{\alpha\beta^2}$, which implies that $\alpha\tau^2\beta^2 < \tau^{-1}$.

We also have

$$\begin{aligned}
G^*(B^T\gamma) - G^*(B^T\lambda^+) &\geq G^*(B^T\lambda^+) + \langle \gamma - \lambda^+, B\nabla G^*(B^T\lambda^+)\rangle - G^*(B^T\lambda^+) \qquad (20)\\
&= \langle \gamma - \lambda^+, \Phi(\lambda^+)\rangle. \qquad (21)
\end{aligned}$$

Adding estimate (18–19) to (20–21) yields

$$\begin{aligned}
D(\lambda^+) - D(\gamma) &= G^*(B^T\gamma) - G^*(B^T\lambda^+) \\
&\quad + H^*(A^T\gamma) - H^*(A^T\lambda^+) \\
&\quad + \langle \lambda^+ - \gamma, b\rangle \\
&\geq \langle \gamma - \lambda^+, \Phi(\lambda^+)\rangle \\
&\quad + \langle \gamma - \lambda^+, \Psi(\lambda^{\frac{1}{2}})\rangle - \frac{1}{2\tau}\|\lambda^+ - \lambda\|^2 \\
&\quad + \langle \lambda^+ - \gamma, b\rangle \\
&= \langle \gamma - \lambda^+, \Phi(\lambda^+) + \Psi(\lambda^{\frac{1}{2}}) - b\rangle - \frac{1}{2\tau}\|\lambda^+ - \lambda\|^2 \qquad (22)\\
&= \tau^{-1}\langle \gamma - \lambda^+, \lambda - \lambda^+\rangle - \frac{1}{2\tau}\|\lambda^+ - \lambda\|^2 \qquad (23)\\
&= \tau^{-1}\langle \gamma - \lambda + \lambda - \lambda^+, \lambda - \lambda^+\rangle - \frac{1}{2\tau}\|\lambda^+ - \lambda\|^2 \\
&= \tau^{-1}\langle \gamma - \lambda, \lambda - \lambda^+\rangle + \frac{1}{2\tau}\|\lambda^+ - \lambda\|^2,
\end{aligned}$$

where we have used Lemma 1 to traverse from (22) to (23).

□

Note that the hypothesis of Lemma 2 is actually a bit stronger than necessary. We actually only need the weaker condition

$$\tau^3 \leq \frac{1}{L(\Psi)L(\Phi)^2}.$$

Often, we do not have direct access to the dual function, let alone the composite functions Ψ and Φ. In this case, the stronger assumption $\tau^3 \leq \frac{\sigma_H\sigma_G^2}{\rho(A^TA)\rho(B^TB)^2}$ is sufficient to guarantee that the results of Lemma 2 hold.

Also, note that Lemma 2 requires $Bv = \Phi(\lambda)$. This can be thought of as a "compatibility condition" between v and λ. While we are considering the action of ADMM on the dual variables, the behavior of the iteration is unpredictable if we allow arbitrary values of v to be used at the start of each iteration. Our bounds on the dual function are only meaningful if a compatible value of v is chosen. This poses no difficulty, because the iterates produced by Algorithm 1 satisfy this condition automatically — indeed Lemma 1 guarantees that for arbitrary (v_k, λ_k) we have $Bv_{k+1} = \Phi(\lambda_{k+1})$. It follows that $Bv_k = \Phi(\lambda_k)$ for all $k > 0$, and so our iterates satisfy the hypothesis of Lemma 2.

3.2. **Convergence Results for ADMM.** We are now ready to prove a global convergence result for non-accelerated ADMM (Algorithm 1) under strong convexity assumptions.

Theorem 1. *Consider the ADMM iteration described by Algorithm 1. Suppose that H and G satisfy the conditions of Assumption 1, and that $\tau^3 \leq \frac{\sigma_H \sigma_G^2}{\rho(A^TA)\rho(B^TB)^2}$. Then for $k > 1$ the sequence of dual variables $\{\lambda_k\}$ satisfies*

$$D(\lambda^\star) - D(\lambda_k) \leq \frac{\|\lambda^\star - \lambda_1\|^2}{2\tau(k-1)},$$

where $\lambda^\star$ is a Lagrange multiplier that maximizes the dual.

Proof. We begin by plugging $\gamma = \lambda^\star$ and $(v, \lambda) = (v_k, \lambda_k)$ into Lemma 2. Note that $\lambda^+ = \lambda_{k+1}$ and

$$2\tau(D(\lambda_{k+1}) - D(\lambda^\star)) \geq \|\lambda_{k+1} - \lambda_k\|^2 + 2\langle \lambda_k - \lambda^\star, \lambda_{k+1} - \lambda_k\rangle \tag{24}$$

$$= \|\lambda^\star - \lambda_{k+1}\|^2 - \|\lambda^\star - \lambda_k\|^2. \tag{25}$$

Summing over $k = 1, 2, \ldots n-1$ yields

$$2\tau\left(-(n-1)D(\lambda^\star) + \sum_{k=1}^{n-1} D(\lambda_{k+1})\right) \geq \|\lambda^\star - \lambda_n\|^2 - \|\lambda^\star - \lambda_1\|^2. \tag{26}$$

Now, we use Lemma 2 again, with $\lambda = \gamma = \lambda_k$ and $v = v_k$, to obtain

$$2\tau\left(D(\lambda_{k+1}) - D(\lambda_k)\right) \geq \|\lambda_k - \lambda_{k+1}\|^2$$

Multiply this estimate by $k-1$, and summing over iterates 1 through $n-1$ gives us

$$2\tau \sum_{k=1}^{n-1} \left(kD(\lambda_{k+1}) - D(\lambda_{k+1}) - (k-1)D(\lambda_k)\right) \geq \sum_{k=1}^{n-1} k\|\lambda_k - \lambda_{k+1}\|^2.$$

The telescoping sum on the left reduces to

$$2\tau\left((n-1)D(\lambda_n) - \sum_{k=1}^{n-1} D(\lambda_{k+1})\right) \geq \sum_{k=1}^{n-1} k\|\lambda_k - \lambda_{k+1}\|^2. \tag{27}$$

Adding (26) to (27) generates the inequality

$$2(n-1)\tau\left(D(\lambda_n) - D(\lambda^\star)\right) \geq \|\lambda^\star - \lambda_n\|^2 - \|\lambda^\star - \lambda_1\|^2 + \sum_{k=1}^{n-1} k\|\lambda_k - \lambda_{k+1}\|^2 \geq -\|\lambda^\star - \lambda_1\|^2,$$

from which we arrive at the bound

$$D(\lambda^\star) - D(\lambda_k) \leq \frac{\|\lambda^\star - \lambda_1\|^2}{2\tau(k-1)}.$$

□

Note that the iterates generated by Algorithm 1 depend on the initial value for both the primal variable v_0 and the dual variable λ_0. However the error bounds in Theorem 1 involve only the dual variable. This is possible, because our error bounds involve the iterate λ_1 rather than the initial value λ_0 chosen by the user. Because the value of λ_1 depends on our choice of v_0, the dependence on the primal variable is made implicit. Note also that for some problems the optimal $\lambda^\star$ may not be unique. For such problems, the above theorem does not guarantee convergence of the sequence $\{\lambda_k\}$ itself. Other analytical approaches can be used to prove convergence of $\{\lambda_k\}$ (see for example [32]), however we use the above techniques because they lend themselves to the analysis of an accelerated ADMM scheme. Strong results will be proved about convergence of the residuals in Section 3.3.

3.3. **Optimality Conditions and Residuals.** Using Theorem 1, it is possible to put global bounds on the size of the primal and dual residuals, as defined in (7) and (8).

Lemma 3. *When Assumption 1 holds, we have the following rates of convergence on the residuals:*

$$\begin{array}{rcl} \|r_k\|^2 & \leq & \mathcal{O}(1/k), \\ \|d_k\|^2 & \leq & \mathcal{O}(1/k). \end{array}$$

Proof. Applying Lemma 2 with $\gamma = \lambda_k$ and $v = v_k$ gives us

$$D(\lambda_{k+1}) - D(\lambda_k) \geq (\tau/2)\|\lambda_k - \lambda_{k+1}\|^2.$$

Rearranging this and combining with the convergence rate of the dual function, we have

$$\mathcal{O}(1/k) \geq D(\lambda^\star) - D(\lambda_k) \geq D(\lambda^\star) - D(\lambda_{k+1}) + (\tau/2)\|\lambda_k - \lambda_{k+1}\| \geq (\tau/2)\|\lambda_k - \lambda_{k+1}\|^2.$$

From the λ update, this implies

$$\|r_k\|^2 = \|b - Au_k - Bv_k\|^2 \leq \mathcal{O}(1/k).$$

Under the assumption that G is strongly convex and that $Bv_k \in \Phi(\lambda_k)$, we have that

$$\|Bv_k - Bv_{k-1}\|^2 = \|\Phi(\lambda_k) - \Phi(\lambda_{k-1})\|^2 \leq \beta^2\|\lambda_k - \lambda_{k-1}\|^2 \leq \mathcal{O}(1/k).$$

This immediately implies that the dual residual norm squared

$$\|d_k\|^2 = \|\tau A^T B(v_k - v_{k-1})\|^2 \leq \mathcal{O}(1/k).$$

□

3.4. **Comparison to Other Results.** It is possible to achieve convergence results with weaker assumptions than those required by Theorem 1 and Lemma 3. The authors of [33] prove $O(1/k)$ convergence for ADMM. While this result does not require any strong convexity assumptions, it relies a fairly unconventional measure of convergence obtained from the variational inequality formulation of (1). The results in [33] are the most general convergence rate results for ADMM to date.

Since the original appearance of this manuscript, several other authors has provided convergence results for ADMM under strong-convexity assumptions. The authors of [32] prove R-linear convergence of ADMM under strong convexity assumptions similar to those used in the present work. The work [32] also provides linear convergence results under the assumption of only a single strongly convex term provided the linear operators A and B are full rank. These convergence results bound the error as measured by an approximation to the primal-dual duality gap.

R-linear convergence of ADMM with an arbitrary number of objective terms is proved in [34] using somewhat stronger assumptions than those of Theorems 1. In particular, it is required that each objective term of (1) have the form $F(Ax) + E(x)$. where F is strictly convex and E is polyhedral. It is also required that all variables in the minimization be restricted to lie in a compact set.

While Theorem 1 and Lemma 3 do not contain the strongest convergence rates available under strong convexity assumptions, they have the advantage that they bound fairly conventional measures of the error (i.e., the dual objective and residuals). Also, the proof of Theorem 1 is instructive in that it lays the foundation for the accelerated convergence results of Section 4.

4. Fast ADMM For Strongly Convex Problems

In this section, we develop the accelerated variant of ADMM listed in Algorithm 7. The accelerated method is simply ADMM with a predictor-corrector type acceleration step. This predictor-corrector step is stable only in the special case where both objective terms in (1) are strongly convex. In Section 4.3 we introduce a restart rule that guarantees convergence for general (i.e., weakly convex) problems.

Algorithm 7 Fast ADMM for Strongly Convex Problems

```
Require: $v_{-1} = \hat{v}_0 \in R^{N_v}, \lambda_{-1} = \hat{\lambda}_0 \in R^{N_b}, \tau > 0$
for $k = 1, 2, 3 \ldots$ do
    $u_k = \text{argmin}\, H(u) + \langle \hat{\lambda}_k, -Au \rangle + \frac{\tau}{2}\|b - Au - B\hat{v}_k\|^2$
    $v_k = \text{argmin}\, G(v) + \langle \hat{\lambda}_k, -Bv \rangle + \frac{\tau}{2}\|b - Au_k - Bv\|^2$
    $\lambda_k = \hat{\lambda}_k + \tau(b - Au_k - Bv_k)$
    if $E_k > 0$ then
        $\alpha_{k+1} = \frac{1+\sqrt{1+4\alpha_k^2}}{2}$
        $\hat{v}_{k+1} = v_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(v_k - v_{k-1})$
        $\hat{\lambda}_{k+1} = \lambda_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\lambda_k - \lambda_{k-1})$
    else
        $\alpha_{k+1} = 1$, $\hat{v}_{k+1} = v_k$, $\hat{\lambda}_{k+1} = \lambda_k$
    end if
end for
```

4.1. **Convergence of Fast ADMM for Strongly Convex Problems.** We consider the convergence of algorithm 7. We shall here consider the case when H and G satisfy Assumption 1 – i.e., both objective terms are strongly convex.

Define

$$s_k = a_k \lambda_k - (a_k - 1)\lambda_{k-1} - \lambda^\star.$$

We will need the following identity:

Lemma 4. *Let $\lambda_k, \hat{\lambda}_k$, and a_k be defined as in Algorithm 7. Then*

$$s_{k+1} = s_k + a_{k+1}(\lambda_{k+1} - \hat{\lambda}_{k+1}).$$

Proof. We begin by noting that, from the definition of $\hat{\lambda}_{k+1}$ given in Algorithm 7,

$$(a_k - 1)(\lambda_k - \lambda_{k-1}) = a_{k+1}(\hat{\lambda}_{k+1} - \lambda_k).$$

This observation, combined with the definition of s_k give us

$$\begin{aligned}
s_{k+1} &= a_{k+1}\lambda_{k+1} - (a_{k+1} - 1)\lambda_k - \lambda^\star \\
&= \lambda_k - \lambda^\star + a_{k+1}(\lambda_{k+1} - \lambda_k) \\
&= \lambda_k - (a_k - 1)\lambda_{k-1} - \lambda^\star + a_{k+1}(\lambda_{k+1} - \lambda_k) + (a_k - 1)\lambda_{k-1} \\
&= a_k\lambda_k - (a_k - 1)\lambda_{k-1} - \lambda^\star + a_{k+1}(\lambda_{k+1} - \lambda_k) - (a_k - 1)(\lambda_k - \lambda_{k-1}) \\
&= s_k + a_{k+1}(\lambda_{k+1} - \lambda_k) - (a_k - 1)(\lambda_k - \lambda_{k-1}) \\
&= s_k + a_{k+1}(\lambda_{k+1} - \lambda_k) - a_{k+1}(\hat{\lambda}_{k+1} - \lambda_k) \\
&= s_k + a_{k+1}(\lambda_{k+1} - \hat{\lambda}_{k+1}).
\end{aligned}$$

□

This identity, along with Lemma 2, we can be used to derive the following relation.

Lemma 5. *Suppose that H and G satisfy Assumption 1, G is quadratic, and $\tau^3 \le \frac{\sigma_H \sigma_G^2}{\rho(A^TA)\rho(B^TB)^2}$. The iterates generated by Algorithm 7 without restart and the sequence $\{s_k\}$ obey the following relation:*

$$\|s_{k+1}\|^2 - \|s_k\|^2 \le 2a_k^2\tau\left(D(\lambda^\star) - D(\lambda_k)\right) - 2a_{k+1}^2\tau\left(D(\lambda^\star) - D(\lambda_{k+1})\right).$$

Proof. See Appendix A. □

Using these preliminary results, we now prove a convergence theorem for accelerated ADMM.

Theorem 2. *Suppose that H and G satisfy Assumption 1 and that G is quadratic. If we choose $\tau^3 \leq \frac{\sigma_H \sigma_G^2}{\rho(A^T A)\rho(B^T B)^2}$, then the iterates $\{\lambda_k\}$ generated by Algorithm 7 without restart satisfy*

$$D(\lambda^\star) - D(\lambda_k) \leq \frac{2\|\hat{\lambda}_1 - \lambda^\star\|^2}{\tau(k+2)^2}.$$

Proof. From Lemma 5 we have

$$\begin{aligned} 2a_{k+1}^2 \tau \left(D(\lambda^\star) - D(\lambda_{k+1})\right) &\leq \|s_k\|^2 - \|s_{k+1}\|^2 \\ &\quad + 2a_k^2 \tau \left(D(\lambda^\star) - D(\lambda_k)\right) \\ &\leq \|s_k\|^2 + 2a_k^2 \tau \left(D(\lambda^\star) - D(\lambda_k)\right). \end{aligned} \tag{28}$$

Now, note that the result of Lemma 5 can be written

$$\|s_{k+1}\|^2 + 2a_{k+1}^2 \tau \left(D(\lambda^\star) - D(\lambda_{k+1})\right) \leq \|s_k\|^2 + 2a_k^2 \tau \left(D(\lambda^\star) - D(\lambda_k)\right),$$

which implies by induction that

$$\|s_k\|^2 + 2a_k^2 \tau \left(D(\lambda^\star) - D(\lambda_k)\right) \leq \|s_1\|^2 + 2a_1^2 \tau \left(D(\lambda^\star) - D(\lambda_1)\right).$$

Furthermore, applying (49) (from Appendix A) with $k = 0$ yields

$$D(\lambda_1) - D(\lambda^\star) \geq \frac{1}{2\tau}\|\lambda_1 - \hat{\lambda}_1\|^2 + \frac{1}{\tau}\langle \hat{\lambda}_1 - \lambda^\star, \lambda_1 - \hat{\lambda}_1\rangle = \frac{1}{2\tau}\left(\|\lambda_1 - \lambda^\star\|^2 - \|\hat{\lambda}_1 - \lambda^\star\|^2\right).$$

Applying these observations to (28) yields

$$\begin{aligned} 2a_{k+1}^2 \tau \left(D(\lambda^\star) - D(\lambda_{k+1})\right) &\leq \|s_1\|^2 + 2\tau \left(D(\lambda^\star) - D(\lambda_1)\right) \\ &= \|\lambda_1 - \lambda^\star\|^2 + 2\tau \left(D(\lambda^\star) - D(\lambda_1)\right) \\ &\leq \|\lambda_1 - \lambda^\star\|^2 + \|\hat{\lambda}_1 - \lambda^\star\|^2 - \|\lambda_1 - \lambda^\star\|^2 \\ &= \|\hat{\lambda}_1 - \lambda^\star\|^2, \end{aligned}$$

from which it follows immediately that

$$D(\lambda^\star) - D(\lambda_k) \leq \frac{\|\hat{\lambda}_1 - \lambda^\star\|^2}{2\tau a_k^2}.$$

It remains only to observe that $a_k > a_{k-1} + \frac{1}{2} > 1 + \frac{k}{2}$, from which we arrive at

$$D(\lambda^\star) - D(\lambda_k) \leq \frac{2\|\hat{\lambda}_1 - \lambda^\star\|^2}{\tau(k+2)^2}.$$

□

4.2. **Convergence of Residuals.** The primal and dual residuals for ADMM, equations (7) and (8), can be extended to measure convergence of the accelerated Algorithm 7. In the accelerated case the primal residual is unchanged, $r_k = b - Au_k - Bv_k$; a simple derivation yields the following new dual residual

$$d_k = \tau A^T B(v_k - \hat{v}_k). \tag{29}$$

We now prove global convergence rates in terms of the primal and dual residuals.

Lemma 6. *If Assumption 1 hold, G is quadratic, and B has full row rank, then the following rates of convergence hold for Algorithm 7:*

$$\begin{aligned} \|r_k\|^2 &\leq \mathcal{O}(1/k^2) \\ \|d_k\|^2 &\leq \mathcal{O}(1/k^2). \end{aligned}$$

Proof. Since G is quadratic and B has full row rank, $-D$ is strongly convex, which implies that

$$\|\lambda^\star - \lambda_k\|^2 \leq \mathcal{O}(1/k^2)$$

from the convergence of $D(\lambda_k)$. This implies that

$$\|\lambda_{k+1} - \lambda_k\|^2 \leq (\|\lambda^\star - \lambda_k\| + \|\lambda^\star - \lambda_{k+1}\|)^2 \leq \mathcal{O}(1/k^2).$$

Since H and G are strongly convex, D has a Lipschitz continuous gradient with constant $L \leq 1/\sigma_H + 1/\sigma_G$. From this we have that

$$D(\hat{\lambda}_k) \geq D(\lambda^\star) - (L/2)\|\hat{\lambda}_k - \lambda^\star\|^2$$

and so

$$\begin{aligned} D(\lambda_{k+1}) - D(\hat{\lambda}_{k+1}) &\leq D(\lambda_{k+1}) - D(\lambda^\star) + (L/2)\|\hat{\lambda}_{k+1} - \lambda^\star\|^2 \\ &\leq (L/2)\|\hat{\lambda}_{k+1} - \lambda^\star\|^2 \\ &= (L/2)\|\lambda_k - \lambda^\star + \gamma_k(\lambda_k - \lambda_{k-1})\|^2 \\ &\leq (L/2)\left(\|\lambda_k - \lambda^\star\| + \gamma_k\|\lambda_k - \lambda_{k-1}\|\right)^2 \\ &\leq \mathcal{O}(1/k^2), \end{aligned}$$

where $\gamma_k = (a_k - 1)/a_{k+1} \leq 1$. But from Lemma 2 we have that

$$D(\lambda_{k+1}) - D(\hat{\lambda}_{k+1}) \geq \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|^2 = (1/2)\|r_k\|^2$$

and therefore

$$\|r_k\|^2 \leq \mathcal{O}(1/k^2).$$

We now consider convergence of the dual residual. From Lemma 8 (see Appendix A) we know that $B\hat{v}_k \in \Phi(\hat{\lambda}_k)$, and $Bv_k \in \Phi(\lambda_k)$. It follows that

$$\begin{aligned} \|d_k\|^2 &= \|\tau A^T B(v_k - \hat{v}_k)\|^2 \leq \|\tau A^T(\Phi(\lambda_k) - \Phi(\hat{\lambda}_k))\|^2 \\ &\leq \tau^2 \rho(A^T A)\beta^2\|\lambda_k - \hat{\lambda}_k\|^2 \leq \mathcal{O}(1/k^2). \end{aligned}$$

□

4.3. **Accelerated ADMM for Weakly Convex Problems.** In this section, we consider the application of ADMM to weakly convex problems. We can still accelerate ADMM in this case, however we cannot guarantee a global convergence rate as in the strongly convex case. For weakly convex problems, we must enforce stability using a restart rule, as described in Algorithm 8.

The restart rule relies on a combined residual, which measures both the primal and dual error simultaneously:

$$c_k = \frac{1}{\tau}\|\lambda_k - \hat{\lambda}_k\|^2 + \tau\|B(v_k - \hat{v}_k)\|^2. \tag{30}$$

Note the combined residual contains two terms. The first term $\frac{1}{\tau}\|\lambda_k - \hat{\lambda}_k\|^2 = \tau\|b - Au_k - Bv_k\|^2$ measures the primal residual (7). The second term $\tau\|B(v_k - \hat{v}_k)\|^2$ is closely related to the dual residual (8) but differs in that is does not require multiplication by A^T.

Algorithm 8 involves a new parameter $\eta \in (0, 1)$. On Line 6 of the algorithm, we test whether the most recent ADMM step has decreased the combined residual by a factor of at least η. If so, then we proceed with the acceleration (Steps 7–9). Otherwise, we throw out the most recent iteration and "restart" the algorithm by setting $\alpha_{k+1} = 1$. In this way, we use restarting to enforce monotonicity of the method. Similar restart rules of this type have been studied for unconstrained minimization in [35]. Because it is desirable to restart the method as infrequently as possible, we recommend a value of η close to 1. In all the numerical experiments presented here, $\eta = 0.999$ was used.

4.4. **Convergence of the Restart Method.** To prove the convergence of Algorithm 8, we will need the following result, due to Yuan and He (for a proof, see [33] Theorem 4.1):

Lemma 7. *The iterates $\{v_k, \lambda_k\}$ produced by Algorithm 1 satisfy*

$$\frac{1}{\tau}\|\lambda_{k+1} - \lambda_k\|^2 + \tau\|B(v_{k+1} - v_k)\|^2 \leq \frac{1}{\tau}\|\lambda_k - \lambda_{k-1}\|^2 + \tau\|B(v_k - v_{k-1})\|^2.$$

In words, Theorem 7 states that the unaccelerated ADMM (Algorithm 1) decreases the combined residual monotonically. It is now fairly straightforward to prove convergence for the restart method.

Algorithm 8 Fast ADMM with Restart

```
Require: $v_{-1} = \hat{v}_0 \in R^{N_v}, \lambda_{-1} = \hat{\lambda}_0 \in R^{N_b}, \tau > 0, \eta \in (0,1)$
 1: for $k = 1, 2, 3 \ldots$ do
 2:     $u_k = \operatorname{argmin} H(u) + \langle \hat{\lambda}_k, -Au \rangle + \frac{\tau}{2}\|b - Au - B\hat{v}_k\|^2$
 3:     $v_k = \operatorname{argmin} G(v) + \langle \hat{\lambda}_k, -Bv \rangle + \frac{\tau}{2}\|b - Au_k - Bv\|^2$
 4:     $\lambda_k = \hat{\lambda}_k + \tau(b - Au_k - Bv_k)$
 5:     $c_k = \tau^{-1}\|\lambda_k - \hat{\lambda}_k\|^2 + \tau\|B(v_k - \hat{v}_k)\|^2$
 6:     if $c_k < \eta c_{k-1}$ then
 7:         $\alpha_{k+1} = \frac{1+\sqrt{1+4\alpha_k^2}}{2}$
 8:         $\hat{v}_{k+1} = v_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(v_k - v_{k-1})$
 9:         $\hat{\lambda}_{k+1} = \lambda_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\lambda_k - \lambda_{k-1})$
10:     else
11:         $\alpha_{k+1} = 1$, $\hat{v}_{k+1} = v_{k-1}$, $\hat{\lambda}_{k+1} = \lambda_{k-1}$
12:         $c_k \leftarrow \eta^{-1} c_{k-1}$
13:     end if
14: end for
```

Theorem 3. *For convex H and G, Algorithm 8 converges in the sense that*

$$\lim_{k\to\infty} c_k = 0.$$

Proof. We begin with some terminology. Each iteration of Algorithm 8 is of one of three types: (1) A "restart" iteration occurs when the inequality in Step 6 of the algorithm is not satisfied. (2) A "non-accelerated" iteration occurs immediately after a "restart" iteration. On such iterations $\alpha_k = 1$, and so the acceleration (Lines 7–9) of Algorithm 8 are inactivated making the iteration equivalent to the original ADMM. (3) An "accelerated" iteration is any iteration that is not "restart" or "unaccelerated." On such iterations, Lines 7–9 of the algorithm are invoked and $\alpha_k > 1$.

Suppose that a restart occurs at iteration k. Then the values of v_k and λ_k are returned to their values at iteration $k-1$ (which have combined residual c_{k-1}). We also set $\alpha_{k+1} = 1$, making iteration $k+1$ an unaccelerated iteration. By Theorem 3 the combined residual is non-increasing on this iteration and so $c_{k+1} \le c_{k-1}$. Note that on accelerated steps, the combined residual decreases by at least a factor of η. It follows that the combined residual satisfies

$$c_k \le c_0 \eta^{\hat{k}},$$

where $\hat{k}$ denotes the number of accelerated steps have occurred within the first k iterations.

Clearly, if the number of accelerated iterations is infinite, then we have $c_k \to 0$ at $k \to \infty$. In the case that the number of accelerated iterations is finite, then after the final accelerated iteration each pair of restart and unaccelerated iterations is equivalent to a single iteration of the original unaccelerated ADMM (Algorithm 1) for which convergence is known [33]. □

While Algorithm 8 enables us to extend accelerated ADMM to weakly convex problems, our theoretical results are weaker in this case because we cannot guarantee a convergence rate as we did in the strongly convex case. Nevertheless, the empirical behavior of the restart method (Algorithm 8) is superior to that of the original ADMM (Algorithm 1), even in the case of strongly convex functions. Similar results have been observed for restarted variants of other accelerated schemes [35].

5. An Accelerated Alternating Minimization Algorithm

We now develop an accelerated version of Algorithm 2, AMA. This acceleration technique is very simple and comes with fairly strong convergence bounds, but at the cost of requiring H to be strongly convex. We begin by considering the convergence of standard unaccelerated AMA. We can prove convergence of this scheme by noting its equivalence to forward-backward splitting on the dual of problem (1), given in (3).

Following the arguments of Tseng [18], we show that the Lagrange multiplier vectors generated by AMA correspond to those produced by performing FBS (Algorithm 4) on the dual. The original proof of Theorem

4 as presented in [18] does not provide a convergence rate, although it relies on techniques similar to those used here.

Theorem 4. *Suppose G are convex, and that H is strongly convex with strong convexity constant σ_H. Then Algorithm 2 converges whenever $\tau < 2\sigma_H/\rho(A^TA)$. Furthermore, the sequence of iterates $\{\lambda_k\}$ are equivalent to those generated by applying FBS to the dual problem.*

Proof. We begin by considering (3), the dual formulation of problem (1). Define the following operators:

$$\Psi(\lambda) = A\nabla H^*(A^T\lambda) \tag{31}$$

$$\Theta(\lambda) = B\partial G^*(B^T\lambda) - b \tag{32}$$

From the optimality condition for Step 3 of Algorithm 2,

$$B^T\lambda_k + \tau B^T(b - Au_{k+1} - Bv_{k+1}) \in \partial G(v_{k+1}),$$

which leads to

$$v_{k+1} \in \partial G^*\left(B^T\lambda_k + \tau B^T(b - Au_{k+1} - Bv_{k+1})\right).$$

Multiplying by B and subtracting b yields

$$Bv_{k+1} - b \in \Theta(\lambda_k + \tau(b - Au_{k+1} - Bv_{k+1})).$$

Next, we multiply by τ and add $\lambda_k + \tau(b - Au_{k+1} - Bv_{k+1})$ to both sides of the above equation to obtain

$$\lambda_k - \tau Au_{k+1} \in (I + \tau\Theta)\left(\lambda_k + \tau(b - Au_{k+1} - Bv_{k+1})\right)$$

which can be re-written as

$$(I + \tau\Theta)^{-1}(\lambda_k - \tau Au_{k+1}) = (\lambda_k + \tau(b - Au_{k+1} - Bv_{k+1})) = \lambda_{k+1}. \tag{33}$$

Note that $(I + \tau\Theta)^{-1}$ is the "resolvent" of the convex function G^*. This inverse is known to be unique and well-defined for arbitrary proper convex functions (see [36]). Next, the optimality condition for Step 2 yields

$$A^T\lambda_k = \nabla H(u_{k+1}) \Leftrightarrow u_{k+1} = \nabla H^*(A^T\lambda_k) \Rightarrow Au_{k+1} = \Psi(\lambda_k). \tag{34}$$

Combining (33) with (34) yields

$$\lambda_{k+1} = (I + \tau\Theta)^{-1}(\lambda_k - \tau\Psi(\lambda_k)) = (I + \tau\Theta)^{-1}(I - \tau\Psi)\lambda_k, \tag{35}$$

which is the forward-backward recursion for the minimization of (3).

This forward-backward splitting converges as long as $\tau < 2/L(\Psi)$. From the definition of Ψ given in (31), we have that $\tau < \frac{2}{L(\partial H^*)\rho(A^TA)}$. If we apply the identity $L(\nabla H^*) = 1/\sigma_H$, we obtain the condition $\tau < 2\sigma_H/\rho(A^TA)$. □

The equivalence between AMA and FBS provides us with a simple method for accelerating the splitting method. By over-relaxing the Lagrange multiplier vectors after each iteration, we obtain an accelerated scheme equivalent to performing FISTA on the dual; see Algorithm 9.

Algorithm 9 Fast AMA

```
Require: $\alpha_0 = 1,\ \lambda_{-1} = \hat{\lambda}_0 \ \in R^{N_b}, \tau < \sigma_H/\rho(A^TA)$
for $k = 0, 1, 2 \ldots$ do
    $u_k = \operatorname{argmin} H(u) + \langle \hat{\lambda}_k, -Au \rangle$
    $v_k = \operatorname{argmin} G(v) + \langle \hat{\lambda}_k, -Bv \rangle + \frac{\tau}{2}\|b - Au_k - Bv\|^2$
    $\lambda_k = \hat{\lambda}_k + \tau(b - Au_k - Bv_k)$
    $\alpha_{k+1} = (1 + \sqrt{1 + 4\alpha_k^2})/2$
    $\hat{\lambda}_{k+1} = \lambda_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\lambda_k - \lambda_{k-1})$
end for
```

Algorithm 9 accelerates Algorithm 2 by exploiting the equivalence between (2) and forward-backward splitting on the dual problem (3). By applying the Nesterov-type acceleration step to the dual variable λ, we obtain an accelerated method which is equivalent to maximizing the dual functional (3) using FISTA. This notion is formalized below.

Theorem 5. *If H is strongly convex and $\tau < \sigma_H/\rho(A^TA)$, then the iterates $\{\lambda_k\}$ generated by the accelerated Algorithm 9 converge in the dual objective with rate $O(1/k^2)$. More precisely, if D denotes the dual objective, then*

$$D(\lambda^\star) - D(\lambda_k) \leq \frac{2\rho(A^TA)\|x_0 - x^\star\|^2}{\sigma_H(k+1)^2}.$$

Proof. We follow the arguments in the proof of Theorem 4, where it was demonstrated that the sequence of Lagrange multipliers $\{\lambda_k\}$ generated by Algorithm 2 is equivalent to the application of forward backward splitting to the dual problem

$$\max_\lambda \Big(D(\lambda) = \Theta(\lambda) + \Psi(\lambda) \Big).$$

By over-relaxing the sequence of iterates with the parameter sequence $\{\alpha_k\}$, we obtain an algorithm equivalent to performing FISTA on the dual objective. The well-known convergence results for FISTA state that the dual error is bounded above by $2L(\Psi)\|x_0 - x^\star\|^2)/(k+1)^2$. If we recall that $L(\Psi) = \rho(A^TA)/\sigma_H$, then we arrive at the stated bound.

□

6. Relation To Other Methods

The methods presented here have close relationships to other accelerated methods. The most immediate relationship is between fast AMA (Algorithm 9) and FISTA, which are equivalent in the sense that AMA is derived by applying FISTA to the dual problem.

A relationship can also be seen between the proposed fast ADMM method, and the accelerated primal-dual scheme of Chambolle and Pock [37]. The latter method addresses the problem

$$\min_u G(Ku) + H(u). \tag{36}$$

The approach of Chambolle and Pock is to dualize one of the terms and solve the equivalent problem

$$\max_\lambda \min_u \langle Ku, \lambda\rangle - G^*(\lambda) + H(u). \tag{37}$$

The steps of the primal-dual method are described below:

$$\begin{aligned} u_{k+1} &= J_{\tau_n \partial H^*}(u_k - \sigma_k K^T \bar{\lambda}_k), \\ \lambda_{k+1} &= J_{\sigma_n \partial G^*}(\lambda_k + \tau_k K x_k), \\ \bar{\lambda}_{k+1} &= \lambda_{k+1} + \alpha(\lambda_{k+1} - \lambda_k), \end{aligned} \tag{38}$$

where α is an over-relaxation parameter.

We wish to show that (38) is equivalent to the ADMM iteration for $K = I$. Applying the ADMM scheme, Algorithm 1, to (36) with $A = -I$, $B = K$, and $b = 0$, yields the sequence of steps

$$\begin{aligned} u_{k+1} &= \operatorname{argmin} H(u) + \langle \lambda_k, u\rangle + \frac{\tau}{2}\|u - Kv_k\|^2, \\ v_{k+1} &= \operatorname{argmin} G(v) + \langle \lambda_k, -Kv\rangle + \frac{\tau}{2}\|u_{k+1} - Kv\|^2, \\ \lambda_{k+1} &= \lambda_k + \tau(u_{k+1} - Kv_{k+1}). \end{aligned} \tag{39}$$

If we further let $K = I$, this method can be expressed using resolvents as

$$\begin{aligned} u_{k+1} &= J_{\tau^{-1}\partial H}(v_k - \tau^{-1}\lambda_k), \\ v_{k+1} &= J_{\tau^{-1}\partial G}(u_{k+1} + \tau^{-1}\lambda_k), \\ \lambda_{k+1} &= \lambda_k + \tau(u_{k+1} - v_{k+1}). \end{aligned} \tag{40}$$

Finally, from Moreau's identity [38], we have

$$J_{\tau^{-1}\partial G}(z) = z - \tau^{-1} J_{\tau \partial G^*}(\tau z).$$

Applying this to (40) simplifies our steps to

$$\begin{aligned} u_{k+1} &= J_{\tau^{-1}\partial H}(v_k - \tau^{-1}\lambda_k), \\ \lambda_{k+1} &= J_{\tau\partial G}(\lambda_k + \tau u_{k+1}), \\ v_{k+1} &= u_{k+1} + (\lambda_k - \lambda_{k+1})/\tau. \end{aligned}$$

Now let $\bar{\lambda}_k = \tau(u_k - v_k) + \lambda_k$. The resulting ADMM scheme is equivalent to (38), with $\sigma_k = 1/\tau_k$.

Thus, in the case $K = I$, both the scheme (38) and Algorithm 8 can also be seen as an acceleration of ADMM. Scheme (38) has a somewhat simpler form than Algorithm 8 and comes with stronger convergence bounds, but at the cost of less generality.

The scheme proposed by Goldfarb in [31] works for problems of the form (36) in the special case that $K = I$. In this sense, Goldfarb's approach is similar to the approach of [37]. However, the method presented in [31] relies on the "symmetric" form of ADMM, which involves two Lagrange multiplier updates per iteration rather than one. An interesting similarity between the method [31] and the fast ADMM (Algorithm 8) is that both schemes test a stability condition before applying acceleration. The method [31], unlike the method presented here, has the advantage of working on problems with two weakly convex objectives, at the cost of requiring $A = B = I$.

7. Numerical Results

In this section, we demonstrate the effectiveness of our proposed acceleration schemes by applying them to a variety of common test problems. We first consider elastic-net regularization. This problem formally satisfies all of the requirements of the convergence theory (e.g. Theorem 2), and thus convergence is guaranteed without restarting the method. We then consider several weakly-convex problems including image denoising, compressed sensing, image deblurring, and finally general quadratic programming (QP).

7.1. Elastic Net Regularization.

It is common in statistics to fit a large number of variables to a relatively small or noisy data set. In such applications, including all of the variables in the regression may result in over-fitting. Variable selection regressions solve this problem by automatically selecting a small subset of the available variables and performing a regression on this active set. One of the most popular variable selection schemes is the "elastic net" regularization, which corresponds to the following regression problem [39]:

$$\min_u \lambda_1 |u| + \frac{\lambda_2}{2}\|u\|^2 + \frac{1}{2}\|Mu - f\|^2, \tag{41}$$

where M represents a matrix of data, f contains the measurements, and u is the vector of regression coefficients.

In the case $\lambda_2 = 0$, problem (41) is often called the Lasso regression, as proposed by Tibshirani [40]. The Lasso regression combines least-squares regression with an ℓ_1 penalty that promotes sparsity of the coefficients, thus achieving the desired variable selection. It has been shown that the above variable selection often performs better when choosing $\lambda_2 > 0$, thus activating the ℓ_2 term in the model. This is because a simple ℓ_1 penalized regression has difficulty selecting groups of variables whose corresponding columns of A are highly correlated. In such situations, simple Lasso regression results in overly sparse solutions, while the elastic net is capable of identifying groups of highly correlated variables [39, 41].

The elastic net is an example of a problem that formally satisfies all of the conditions of Theorem 2, and so restart is not necessary for convergence. To test the behavior of fast ADMM (Algorithm 7) on this problem, we use an example proposed by the authors of [39]. These authors propose to choose 3 random normal vectors v_1, v_2, and v_3 of length 50. We then define M to be the 50×40 matrix

$$M_i = \begin{cases} v_1 + e_i, & i = 1 \dots 5 \\ v_2 + e_i, & i = 6 \dots 10 \\ v_3 + e_i, & i = 11 \dots 15 \\ N(0,1), & i = 16 \dots 40 \end{cases}$$

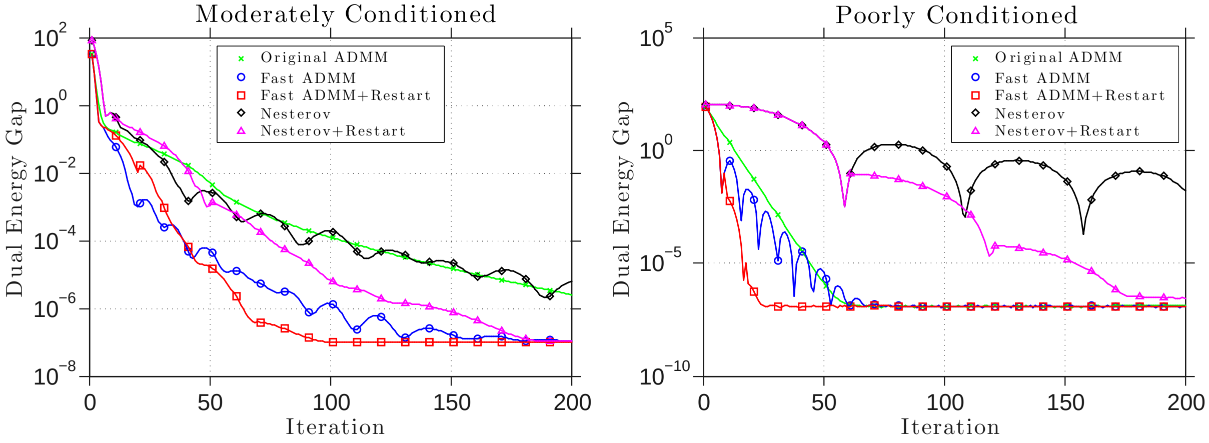


FIGURE 1. Convergence curves for the elastic net problem.

where e_i are random normal vectors with variance σ_e. The problem is then to recover the vector

$$\hat{u} = \begin{cases} 3, & i = 1 \dots 15 \\ 0, & \text{otherwise} \end{cases}$$

from the noisy measurements $f = M\hat{u} + \eta$, where η is normally distributed with standard deviation 0.1. This vector is easily recovered using the model (41) with $\lambda_1 = \lambda_2 = 1$. This test problem simulates the ability of the elastic net to recover a sparse vector from a sensing matrix containing a combination of highly correlated and nearly orthogonal columns.

We apply the fast ADMM method to this problem with $H = \frac{1}{2}\|Mu - f\|^2$, $G = \lambda_1|u| + \frac{\lambda_2}{2}\|u\|^2$, $A = I$, $B = -I$, and $b = 0$. We perform two different experiments. In the first experiment, we choose $\sigma_e = 1$ as suggested by the authors of [39]. This produces a moderately conditioned problem in which the condition number of M is approximately 26. We also consider the case $\sigma_e = 0.1$ which makes the correlations between the first 15 columns of M of stronger, resulting in a condition number of approximately 226.

In both experiments, we choose τ according to Theorem 2, where σ_H is the minimal eigenvalue of $M^T M$ and $\sigma_G = \lambda_2$. Note that Theorem 2 guarantees convergence without using a restart.

We compare the performance of 5 different algorithms on each test problem. We apply the original ADMM (Algorithm 1), the fast ADMM (Algorithm 7), and fast ADMM with restart (Algorithm 8). Because problem (41) is strongly convex, it's dual is Lipschitz differentiable. For this reason, it is possible to solve the dual of (41) using Nesterov's gradient method (Algorithm 5). We consider two variants of Nesterov's method — The original accelerated method (Algorithm 5), and also a variant of Nesterov's method incorporating a restart whenever the method becomes non-monotone [35].

Figure 1 shows sample convergence curves for the experiments described above. The dual energy gap (i.e., $D(\lambda^\star) - D(\lambda_k)$) is plotted for each iterate. The dual energy is a natural measure of error, because the Nesterov method is acting on the dual problem. When $\sigma_e = 1$ the original ADMM without acceleration showed similar performance to both variants of Nesterov's method, and the accelerated ADMM's were somewhat faster. For the poorly-conditioned problem ($\sigma_e = 0.1$) both variants of Nesterov's method performed poorly, and the accelerated ADMM with restart out-performed the other methods by a considerable margin. All five methods had similar iteration costs, with all schemes requiring between 0.24 and 0.28 seconds to complete 200 iterations.

One reason for the superiority of the fast ADMM over Nesterov's method is the stepsize restriction. For this problem, the stepsize restriction for fast ADMM is $\tau^3 \leq \lambda_2^2 \sigma_H$, where σ_H is the minimal eigenvalue of $M^T M$. The stepsize restriction for Nesterov's method is $\tau \leq (\lambda_2 + \sigma_H^{-1})^{-1}$, which is considerably more restrictive when M is poorly conditioned and σ_H is small. For the moderately conditioned problem ADMM is stable with $\tau \leq 0.906$ while Nesterov's method requires $\tau \leq 0.213$. For the poorly conditioned problem, fast ADMM required $\tau \leq 0.201$, while Nesterov's method requires $\tau \leq 0.0041$.

7.2. **Image Restoration.** A fundamental image processing problem is image restoration/denoising. A common image restoration model is the well known total variation (TV) or Rudin-Osher-Fatemi (ROF) [8] model:

$$\text{minimize} \quad |\nabla u| + \tfrac{\mu}{2}\|u - f\|^2. \tag{42}$$

Note that we used the discrete forward gradient operator, $\nabla : R^\Omega \to R^\Omega \times R^2$:

$$(\nabla u)_{i,j} = (u_{i+1,j} - u_{i,j}, u_{i,j+1} - u_{i,j})$$

to denote the discrete total variation of u as simply $|\nabla u|$. When f represents a noisy image, minimizing (42) corresponds to finding a denoised image u that is similar to the noisy image f while being "smooth" in the sense that it has small total variation. The parameter μ controls the tradeoff between the smoothness term and the fidelity term.

To demonstrate the performance of the acceleration scheme, we use three common test images: "cameraman," "Barbara," and "shapes." Images were scaled with pixel values in the 0–255 range and then contaminated with Gaussian white noise of standard deviation $\sigma = 20$. To put the problem in the form of (1), we take $H(u) = \frac{\mu}{2}\|u - f\|^2$, $G(v) = |v|$, $A = \nabla$, and $B = -I$, and $b = 0$. We then have the formulation

$$\begin{array}{ll} \text{minimize} & \frac{\mu}{2}\|u - f\|^2 + |v| \\ \text{subject to} & \nabla u - v = 0. \end{array}$$

over $u \in R^{N_u}$ and $v \in R^{N_v}$. For this problem, we have $\sigma_H = \mu$, and $\rho(A^TA) = \rho(\Delta^T\Delta) < 8$. The stepsize restriction for Algorithm 9 is thus $\tau < \mu/8$.

Using this decomposition, Steps 2 and 3 of Algorithm 9 become

$$\begin{aligned} u_k &= f - \frac{1}{\mu}(\nabla \cdot \hat{\lambda}_k), \\ v_k &= \textbf{shrink}(\nabla u_k + \tau\hat{\lambda}_k, 1/\tau) \end{aligned} \tag{43}$$

where the "shrink" operator is defined in (2). Note that both the u and v update rules involve only explicit vector operations including difference operators (for the gradient and divergence terms), addition, and scalar multiplication. For this reason, the AMA approach for image denoising is easily implemented in numerical packages (such as Matlab), which is one of its primary advantages.

For comparison, we also implement the fast ADMM method with restart using $\tau = \mu/2$, which is the stepsize suggested for this problem in [16]. To guarantee convergence of the method, the first minimization sub-step must be solved exactly, which can be done using the Fast-Fourier Transform (FFT)[16]. While the ADMM methods allows for larger stepsizes than AMA, the iteration cost is higher due to the use of the FFT rather than simple differencing operators.

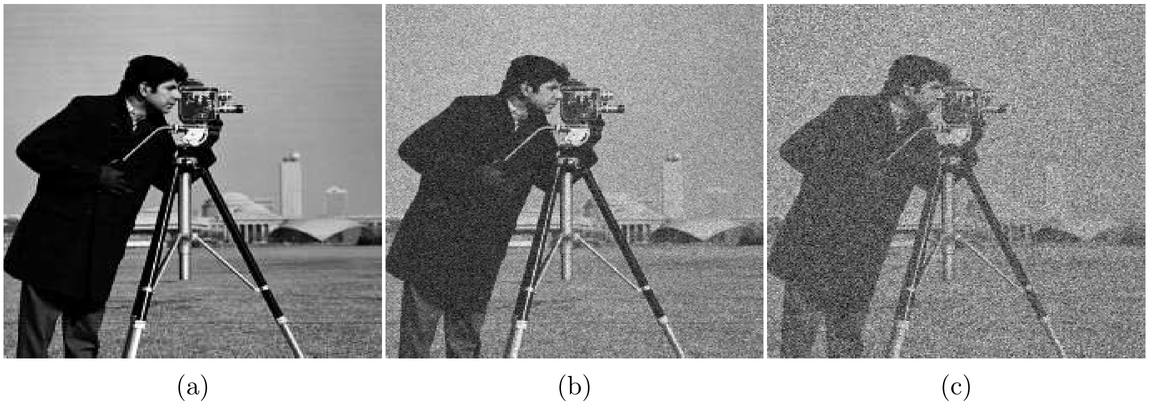
Experiments were conducted on 3 different test images: "cameraman," "Barbara," and "shapes." Images were scaled from 0–255 and a Gaussian noise was added with$\sigma = 20$ and $\sigma = 50$. Images were then denoised with $\mu = 0.1$, 0.05 and $\mu = 0.01$. Sample noise-contaminated images are shown in Figure 2. Denoising results for the case $\mu = 20$ are shown in Figure 3. Iterations were run for each method until they produced an iterate that satisfied $\|u_k - u^\star\|/\|u^\star\| < 0.005$. Below each image we display the un-accelerated/accelerated iteration count for AMA (top) and ADMM (bottom).

A more detailed comparison of the methods is displayed in Table 1, which contains iteration counts and runtimes for both AMA and ADMM. Note that ADMM was advantageous for very course scale problems (small μ), while AMA was more effective for large values of μ. This is expected, because the AMA implementation uses only first-order difference operators which moves information through the image slowly, while ADMM uses the Fourier transforms which move information globally and resolves large smooth regions quickly.

Note also that all four method were considerably slower for smaller μ. However, this effect is much less dramatic when accelerated methods are used.

7.3. **Fast Split Bregman Method for Compressed Sensing and Deblurring.** In this section, we consider the reconstruction of images from measurements in the Fourier domain. A common variational model for such a problem is

$$\text{minimize} \quad |\nabla u| + \tfrac{\epsilon}{2}\|\nabla u\|^2 + \tfrac{\mu}{2}\|R\mathcal{F}u - f\|^2 \tag{44}$$



(a) (b) (c)

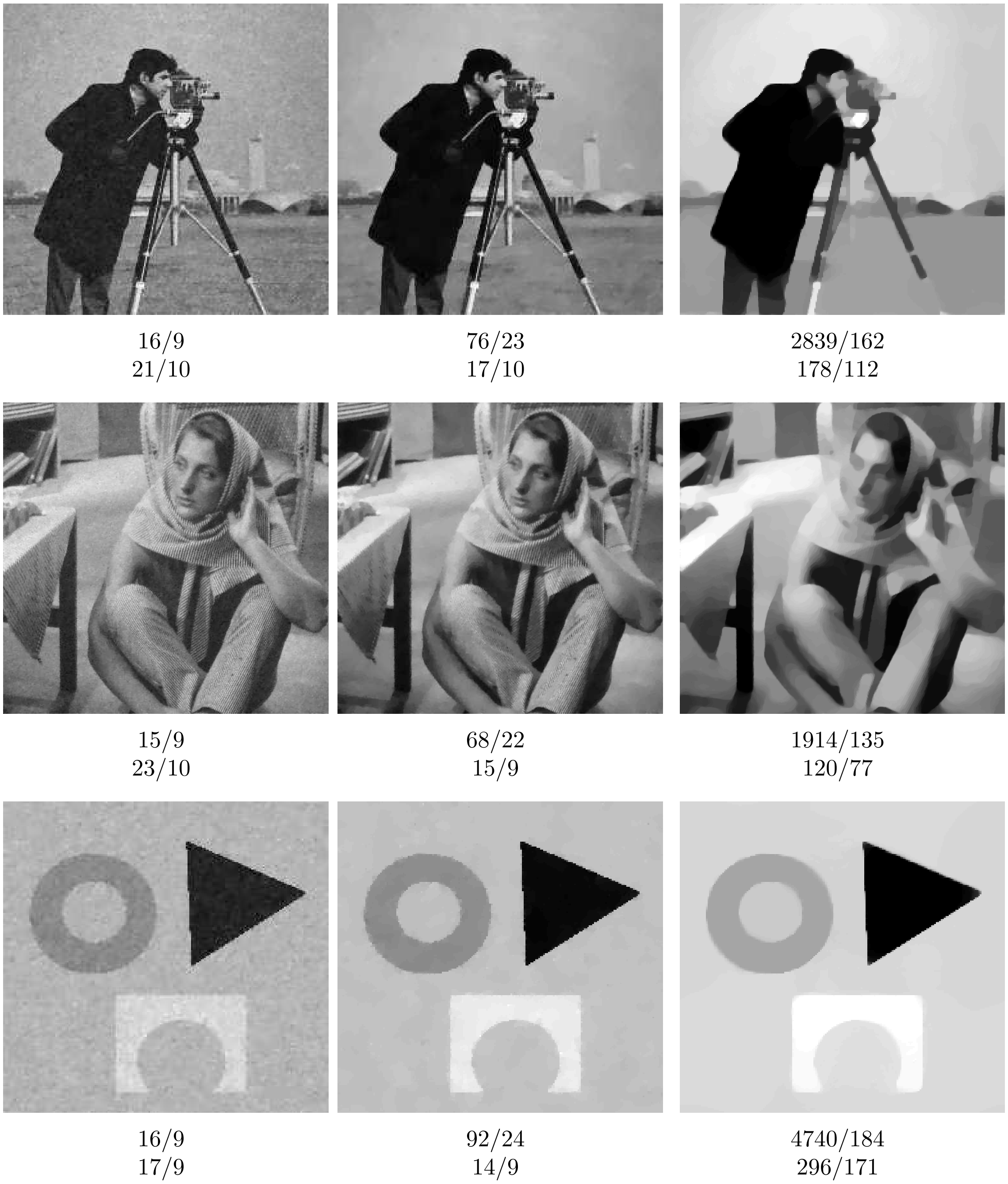
Figure 2. “Cameraman” test image. (a) Original image. (b) Gaussian noise, $\sigma = 20$. (c) Gaussian noise, $\sigma = 50$.

Table 1. Time Trial Results. Iteration counts are reported for each problem/algorithm, with total runtime (sec) in parenthesis.

Image	μ	σ	AMA	Fast AMA	ADMM	Fast ADMM +Restart
Cameraman	0.1	20	16 (0.117)	9 (0.069)	21 (0.293)	10 (0.140)
Cameraman	0.1	50	7 (0.053)	6 (0.049)	37 (0.531)	17 (0.244)
Cameraman	0.05	20	76 (0.550)	23 (0.174)	17 (0.234)	10 (0.139)
Cameraman	0.05	50	24 (0.180)	12 (0.094)	27 (0.382)	15 (0.213)
Cameraman	0.01	20	2839 (20.601)	162 (1.213)	178 (2.381)	112 (1.499)
Cameraman	0.01	50	1814 (13.083)	123 (0.925)	114 (1.530)	74 (0.993)
Barbara	0.1	20	15 (0.548)	9 (0.353)	23 (1.509)	10 (0.653)
Barbara	0.1	50	7 (0.267)	6 (0.240)	38 (2.515)	17 (1.120)
Barbara	0.05	20	68 (2.567)	22 (0.880)	15 (1.073)	9 (0.658)
Barbara	0.05	50	24 (0.901)	12 (0.476)	28 (1.849)	16 (1.064)
Barbara	0.01	20	1914 (69.039)	135 (5.089)	120 (7.636)	77 (4.894)
Barbara	0.01	50	1345 (48.348)	107 (4.057)	85 (5.385)	56 (3.535)
Shapes	0.1	20	16 (0.071)	9 (0.042)	17 (0.175)	9 (0.093)
Shapes	0.1	50	7 (0.031)	6 (0.028)	36 (0.355)	16 (0.162)
Shapes	0.05	20	92 (0.407)	24 (0.115)	14 (0.149)	9 (0.091)
Shapes	0.05	50	23 (0.102)	12 (0.057)	26 (0.261)	14 (0.142)
Shapes	0.01	20	4740 (20.729)	184 (0.828)	296 (2.754)	171 (1.599)
Shapes	0.01	50	2381 (10.434)	138 (0.618)	149 (1.396)	83 (0.782)

over $u \in R^{N_u}$, where $\mathcal{F}$ denotes the Discrete Fourier Transform, R is a diagonal matrix, and f represents the Fourier information that we have obtained. The data term on the right enforces that the Fourier transform of the reconstructed image be compatible with the measured data, while the total-variation term on the left enforces image smoothness. The parameter μ mediates the tradeoff between these two objectives. We also include an ℓ_2 regularization term with parameter ϵ to make the solution smoother, and to make the fast ADMM more effective. We consider two applications for this type of formulation: image deconvolution and compressed sensing.

The reconstruction of an image from a subset of its Fourier modes is at the basis of Magnetic Resonance Imaging (MRI) [42, 43]. In classical MRI, images are measured in the Fourier domain. Reconstruction consists simply of applying the inverse Fast Fourier Transform. Renewed interest in this area has been seen with the introduction of compressed sensing, which seeks to reconstruct high resolution images from a small number of samples [44, 45]. Within the context of MRI, it has been shown that high-resolution images can be reconstructed from under-sampled information in the Fourier domain [46]. This is accomplished by



16/9
21/10

76/23
17/10

2839/162
178/112

15/9
23/10

68/22
15/9

1914/135
120/77

16/9
17/9

92/24
14/9

4740/184
296/171

Figure 3. Iteration count vs. image scale. Images were denoised with $\mu = 0.1$ (left), $\mu = 0.05$ (center), and $\mu = 0.01$ (right) to achieve varying levels of coarseness. The number of iterations required to reach a relative error tolerance of 5×10^{-3} is reported below each image for both the fast/slow methods with AMA on top and ADMM below.

leveraging the sparsity of images – i.e., the reconstructed image should have a sparse representation. This imaging problem is modeled by (44) if we choose R to be a diagonal "row selector matrix." This matrix has a 1 along the diagonal at entries corresponding to Fourier modes that have been measured, and 0's for Fourier modes that are unknown. The known Fourier data is placed in the vector f.

The next application we consider is image deblurring. In many imaging applications, we wish to obtain an image u form its blurred representation $\tilde{u} = K * u$, where K represents a blurring kernel, and '$*$' is the convolution operator. A common formulation of this problem using total-variation minimization solves the following [47, 48]:

$$\min |\nabla u| + \frac{\epsilon}{2}\|\nabla u\|^2 + \frac{\mu}{2}\|K * u - f\|^2. \tag{45}$$

Where ϵ is an ℓ_2 smoothing parameter. It is well known that linear convolution operators are diagonal in the Fourier domain; that it, the convolution operator can be written as $K * u = \mathcal{F}^T R \mathcal{F} u$ for some diagonal matrix R. If we further note that the FFT is unitary, we can write (45) in the form (44) where R is the Fourier representation of the blur operator and $f = \mathcal{F}\tilde{u}$.

The ADMM is particularly useful for problems of the form (44) because each sub-problem of the method is very easily solved in closed form. For this reason ADMM approaches for this class of problems are common in the image processing literature, where they are sometimes referred to as the Split Bregman method [16, 49]. To apply Algorithm 1 to (44) we let $H(u) = \frac{\mu}{2}\|R\mathcal{F}u - f\|^2$, $G(v) = |v| + \frac{\epsilon}{2}\|v\|^2$, $A = \nabla$, and $B = -I$. The resulting minimization step for v is

$$v_k = \underset{v}{\operatorname{argmin}} \left(|v| + \frac{\epsilon}{2}\|v\|^2 + \langle \lambda_k, v \rangle + \frac{\tau}{2}\|v - \nabla u_k\|^2 \right).$$

This regularized problem can be solved using a modified form of equation (43):

$$v_k = \mathbf{shrink}\left(\frac{\tau}{\tau + \epsilon}(\nabla u_k + \tau \hat{\lambda}_k), \frac{1}{\tau + \epsilon} \right).$$

The minimization for u is more complicated. The optimality condition for this minimization is

$$(\tau\Delta + \mu \mathcal{F}^T R' R \mathcal{F}) u_{k+1} = \mu \mathcal{F}^T R^T f + \nabla \cdot (\lambda_k + \tau v_k).$$

To solve this, we note that the discrete Laplacian operator Δ is itself a convolution and can be written in the form $\Delta = \mathcal{F}^T L \mathcal{F}$, where L is a diagonal matrix. This observation allows us to write the optimality condition for u in the form

$$\mathcal{F}^T(\tau L + \mu R' R)\mathcal{F} u_{k+1} = \mu \mathcal{F}^T R^T f + \nabla \cdot (\lambda_k + \tau v_k).$$

The solution to this matrix equation is then

$$u_{k+1} = \mathcal{F}^T(\tau L + \mu R' R)^{-1} \mathcal{F}\left(\mu \mathcal{F}^T R^T f + \nabla \cdot (\lambda_k + \tau v_k) \right).$$

Note that this solution can be evaluated using only two Fast Fourier Transforms.

We apply Algorithms 1 and 8 to compressed sensing and deblurring problems to examine their performance.

We first consider two compressed sensing problems. The first problem is to reconstruct a synthetic image, the digital Shepp-Logan phantom [50], from under-sampled data. For this purpose, we use a 256 × 256 discretization of the Shepp-Logan phantom. A subset of 25% of the Fourier coefficients is measured at random. The second compressed sensing problem is the reconstruction of a real MR image of a saline phantom with dimensions 741 × 784. Only 50% of the coefficients were used for this example. Both test images and their reconstructions are displayed in Figure 4.

To simplify parameter choice, both images were scaled so that pixel intensities range from zero to one. With this scaling, it was found that a stepsize of $\tau = 2$, regularization parameter of $\epsilon = 1$, and fidelity parameter $\mu = 500$ was effective for both examples. Both the original and fast ADMM methods were applied with the same value of $\epsilon = 1$.

Sample convergence curves are displayed in Figure 5. Note that the acceleration becomes more effective as the algorithm progresses. The original and fast methods have similar performance for approximately 10 iterations of the method, after which the superiority of the accelerated scheme becomes apparent. In Figure 6, the 35th iterates of both algorithms are displayed. Note that the accelerated method resolves large, smooth regions more quickly.

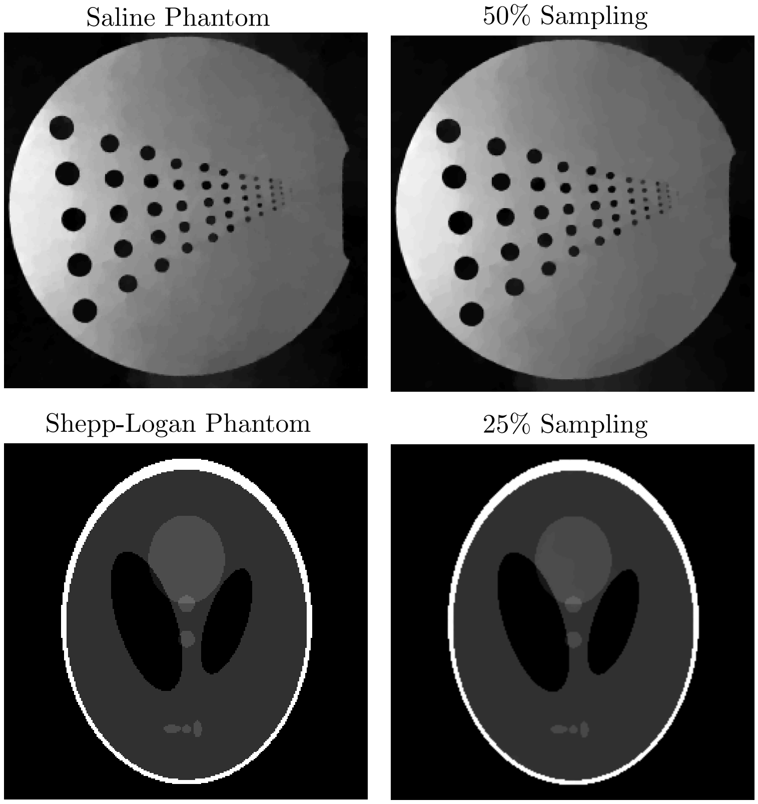


Figure 4. Test images used for compressed sensing experiments (left) and their reconstructions from under-sampled data (right). The saline phantom is displayed on top, with the digital Shepp-Logan phantom on the bottom.

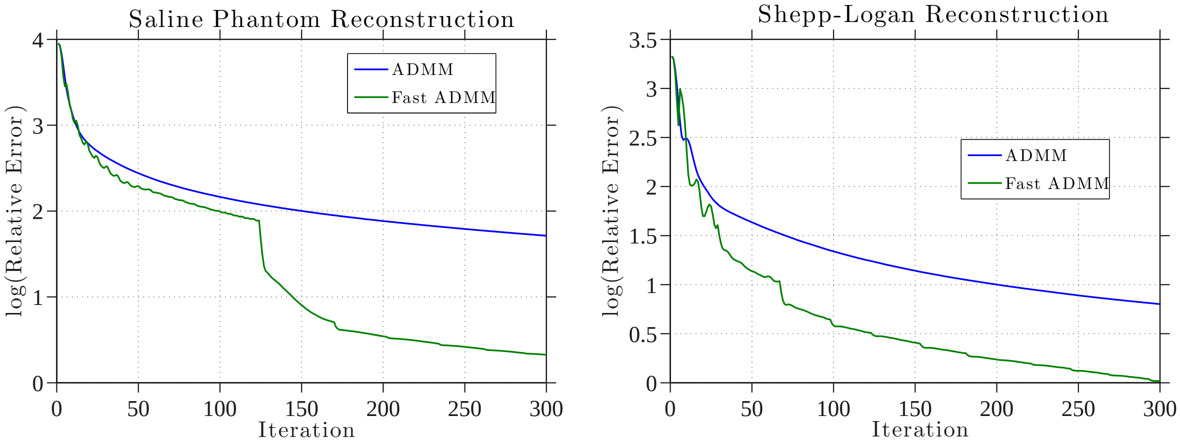


Figure 5. Sample convergence curves showing the performance of the original and fast ADMM methods for the compressed sensing problems. The "relative error" is defined as the sub-optimality of the iterate, i.e., $F(u_k) - F(u^\star)$, where F is the objective function 44.

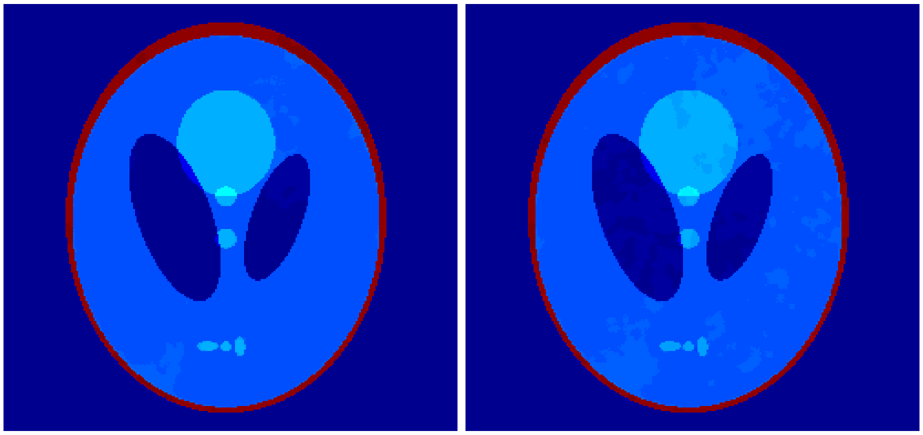


FIGURE 6. False color images showing the 35th iterate of reconstruction of the Shepp- Logan phantom using (left) accelerated ADMM, and (right) standard ADMM.

Two sample deconvolution problems were generated using the “cameraman” and “shapes” test images. Both images were scaled from 0–255. The cameraman and shapes images were blurred by convolving with a Gaussian kernel of radius 2 and 5 pixels, respectively, and contaminated with Gaussian noise of standard deviation 5. The original, blurred, and recovered images are displayed in Figure 7. Deblurring was performed using the formulation (44) with parameters $\mu = 1000$, $\tau = 100$, and $\epsilon = 0$. Note that the ℓ_2 regularization term was deactivated by setting $\epsilon = 0$, as we found that choosing $\epsilon > 0$ resulted in lower quality reconstructions for deblurring problems.

Sample convergence curves for deconvolution problems are shown in Figure 8. Note that the acceleration is less effective for this problem because we were not able to use the ℓ_2 regularization. Nonetheless, the acceleration did yield an improved rate of convergence without any significant computational overhead.

7.4. **General Quadratic Programming.**

Consider the following quadratic program (QP):

$$\begin{array}{ll} \text{minimize} & (1/2)u^TQu + q^Tu \\ \text{subject to} & Au \le b, \end{array} \tag{46}$$

over variable $u \in \mathbf{R}^{N_u}$, where $A \in \mathbf{R}^{N_b \times N_u}$ and $b \in \mathbf{R}^{N_b}$ and where $Q \in \mathbf{R}^{N_u \times N_u}$ is symmetric positive definite. This QP can be solved by both ADMM and AMA; here we compare the performance of standard ADMM and AMA (Algorithms 1 and 2) and their accelerated counterparts (Algorithms 7 and 9) . To transform (46) into the canonical form given by (1), we introduce a new variable $v \in \mathbf{R}^{N_b}$, and rewrite the problem as

$$\begin{array}{ll} \text{minimize} & (1/2)u^TQu + q^Tu + \mathcal{I}_{v \le b}(v) \\ \text{subject to} & Au - v = 0, \end{array} \tag{47}$$

where

$$\mathcal{I}_{v \le b}(v) = \begin{cases} 0 & v \le b \\ \infty & \text{otherwise.} \end{cases}$$

This fits the form of (1) if we set $H(u) = (1/2)u^TQu + q^Tu$ and $G(v) = \mathcal{I}_{v \le b}(v)$.

7.4.1. *ADMM.* With this formulation ADMM is carried out as in Algorithm 10.

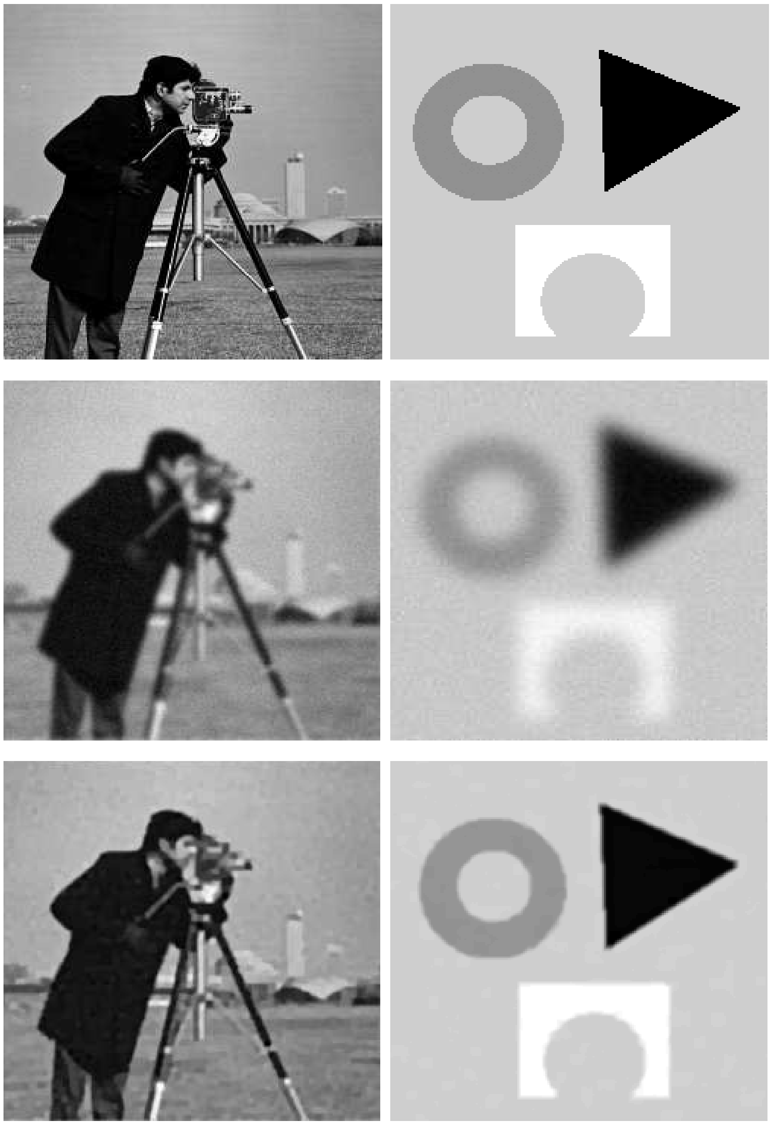


Figure 7. (top) "Cameraman" and "shapes" test images. (center) Blurred test images. (bottom) Restored/deblurred images.

Algorithm 10 ADMM for the QP

```
Require: $v_0 \in \mathbf{R}^{N_v}$, $\lambda_0 \in \mathbf{R}^{N_b}$, $\tau \in \mathbf{R}_+$
for $k = 0, 1, \ldots$ do
    $u_{k+1} = (Q + \tau A^T A)^{-1}(A^T(\lambda_k + \tau v_k) - q)$
    $v_{k+1} = \min(Au_{k+1} - \lambda_k/\tau, b)$
    $\lambda_{k+1} = \lambda_k + \tau(-Au_{k+1} + v_{k+1})$
end for
```

The accelerated form of ADMM follows immediately.

As a numerical instance we took $N_u = 500$ and $N_b = 250$, all quantities were generated randomly, and the condition number of Q was 10^8. At the solution 126 of the constraints were active. Figure 9(a) shows the convergence of the primal and dual residuals for both the unaccelerated and accelerated methods when

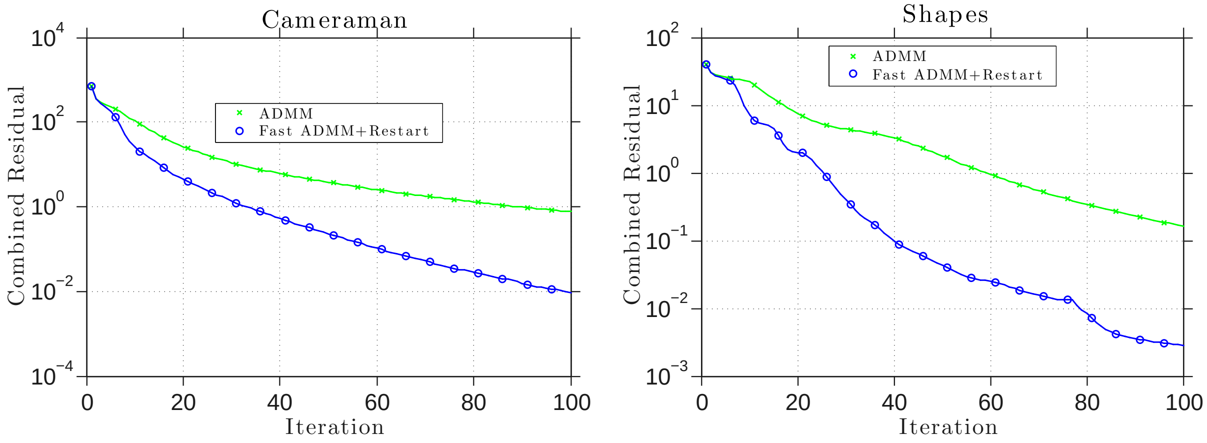
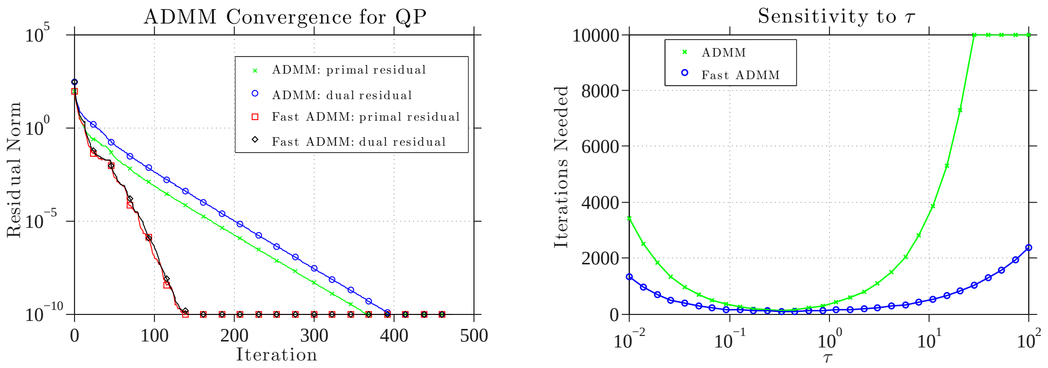


FIGURE 8. Sample convergence curves showing the performance of the original and fast ADMM methods for deblurring problems.

$\tau = 1$. Figure 9(b) shows the number of iterations required to reach a maximum residual norm of 10^{-9} for different choices of τ. We see that the accelerated-restarted form of ADMM requires fewer iterations for all choices of τ and is less sensitive to this choice.



(a) Convergence of primal and dual residuals.

(b) Sensitivity to the choice of τ.

7.4.2. *AMA.* In general it seems that AMA does not perform as well as ADMM when solving QPs. We use it here to illustrate the improvement afforded by accelerating the technique. With the formulation (47), AMA is performed as in Algorithm 11.

Algorithm 11 AMA to solve QP

```
Require: $\lambda_0 \in \mathbf{R}^{N_b}$, $\tau \in \mathbf{R}_+$
for $k = 0, 1, \ldots$ do
    $u_{k+1} = Q^{-1}(A^T\lambda_k - q)$
    $v_{k+1} = \min(Au_{k+1} - \lambda_k/\tau, b)$
    $\lambda_{k+1} = \lambda_k + \tau(-Au_{k+1} + v_{k+1})$
end for
```

The accelerated form follows immediately.

Since H is strongly convex we expect $\mathcal{O}(1/k^2)$ convergence of the dual function D. As a particular example we took $n = 50$ and $m = 25$, again we generated all quantities randomly. The condition number of Q was 4×10^4. At the optimum 12 of the constraints were active; we chose the step size to be $\tau = \lambda_{\min}(Q)/\rho(A^TA)$.

Figure 9 shows the convergence of the dual function for standard AMA, accelerated AMA, and accelerated AMA with restart. The restart rule we chose enforced monotonicity on the dual function iterates, *i.e.*, we restart whenever we have $D(\lambda_k) < D(\lambda_{k-1})$, this is one of the restart rules discussed in [35].

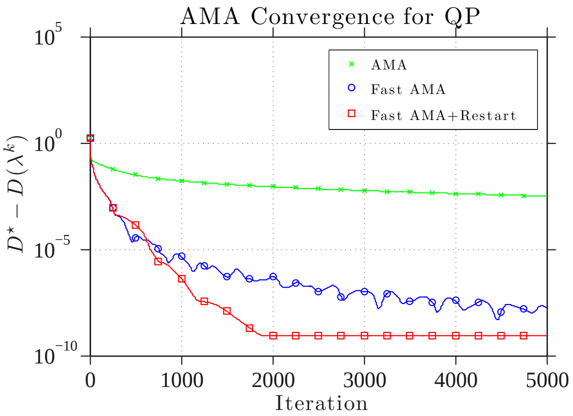


Figure 9. Convergence of the dual function for the QP under AMA.

8. Acknowledgments

This work was possible through the generous support of the National Science Foundation (grant number 08-582).

Thanks to Ernie Esser and Stephen Boyd for many useful discussions.

Appendix A. Proof of Lemma 5

In what follows, we will apply Lemma 2. To do this, we will need the following result stating when the compatibility condition in the hypothesis of Lemma 2 is satisfied.

Lemma 8. *Suppose that H and G satisfy Assumption 1 and that G is quadratic. Then the iterates $\{\hat{v}_k\}$ and $\{\hat{\lambda}_k\}$ satisfy $B\hat{v}_k = \Phi(\hat{\lambda}_k)$ for $k \geq 0$.*

Proof. Note that, by the result of Lemma 1, we have $Bv_k = \Phi(\lambda_k)$. Since we have assumed G to be quadratic and strongly convex, we know that G^*, is quadratic. It follows that ∇G^* is an affine transformation. We then have $B\hat{v}_{k+1} = Bv_k + \frac{\alpha_k - 1}{a_{k+1}}(Bv_k - Bv_{k-1}) \in \Phi(\lambda_k) + \frac{\alpha_k - 1}{a_{k+1}}(\Phi\lambda_k - \Phi\lambda_{k-1}) = \Phi(\hat{\lambda}_{k+1})$ □

Using the above identity, we can prove Lemma 5 from Section 4.1.

Lemma 5 *The iterates generated by Algorithm 7 and the sequence $\{s_k\}$ obey the following relation:*

$$\|s_{k+1}\|^2 - \|s_k\|^2 \leq 2a_k^2\tau\left(D(\lambda^\star) - D(\lambda_k)\right) - 2a_{k+1}^2\tau\left(D(\lambda^\star) - D(\lambda_{k+1})\right)$$

Proof. From Lemma 4, we have

$$\begin{aligned}\|s_{k+1}\|^2 &= \|s_k + a_{k+1}(\lambda_{k+1} - \hat{\lambda}_{k+1})\|^2 \\ &= \|s_k\|^2 + 2a_{k+1}\langle s_k, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle \\ &\quad + a_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|^2.\end{aligned}$$

Rearranging terms yields

$$\begin{aligned}
\|s_{k+1}\|^2 - \|s_k\|^2 &= 2a_{k+1}\langle \lambda_k, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle + a_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|^2 \\
&= 2a_{k+1}\langle a_k\lambda_k - (a_k - 1)\lambda_{k-1} - \lambda^\star, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle \\
&\quad + a_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|^2 \\
&= 2a_{k+1}\langle a_{k+1}\hat{\lambda}_{k+1} + (1 - a_{k+1})\lambda_k - \lambda^\star, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle \\
&\quad + a_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|^2 \\
&= 2a_{k+1}\langle (a_{k+1} - 1)(\hat{\lambda}_{k+1} - \lambda_k) + \hat{\lambda}_{k+1} - \lambda^\star, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle \\
&\quad + a_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|^2 \\
&= 2a_{k+1}(a_{k+1} - 1)\langle \hat{\lambda}_{k+1} - \lambda_k, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle \\
&\quad + 2a_{k+1}\langle \hat{\lambda}_{k+1} - \lambda^\star, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle + a_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|^2 \\
&= 2a_{k+1}(a_{k+1} - 1)\left(\langle \hat{\lambda}_{k+1} - \lambda_k, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle + \frac{1}{2}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|^2\right) \\
&\quad + 2a_{k+1}\left(\langle \hat{\lambda}_{k+1} - \lambda^\star, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle + \frac{1}{2}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|^2\right).
\end{aligned}$$

Note that Lemma 8 guarantees that the iterates of Algorithm 7 satisfy the compatibility condition $B\hat{v}_k = \Phi(\hat{\lambda}_k)$. It follows that the hypothesis of of lemma 2 are satisfied. We now apply Lemma 2 with $\gamma = \lambda_k, \lambda = \hat{\lambda}_{k+1}, v = \hat{v}_{k+1}$. Note that by the notation used in the lemma, $\lambda^+ = \lambda_{k+1}$. This gives us the following bound:

$$D(\lambda_{k+1}) - D(\lambda_k) \geq \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|^2 + \tau^{-1}\langle \hat{\lambda}_{k+1} - \lambda_k, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle. \tag{48}$$

Applying Lemma 2 again with $\gamma = \lambda^\star, \lambda = \hat{\lambda}_{k+1}$, and $v = \hat{v}_{k+1}$ yields

$$D(\lambda_{k+1}) - D(\lambda^\star) \geq \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|^2 + \tau^{-1}\langle \hat{\lambda}_{k+1} - \lambda^*, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle. \tag{49}$$

Applying the estimates (48) and (49) to the above equality gives us

$$\begin{aligned}
\|s_{k+1}\|^2 - \|s_k\|^2 &\leq 2a_{k+1}(a_{k+1} - 1)\tau\left(D(\lambda_{k+1}) - D(\lambda_k)\right) \\
&\quad + 2a_{k+1}\tau\left(D(\lambda_{k+1}) - D(\lambda^\star)\right) \\
&= 2a_{k+1}^2\tau D(\lambda_{k+1}) - 2a_{k+1}(a_{k+1} - 1)\tau D(\lambda_k) \\
&\quad - 2a_{k+1}\tau D(\lambda^\star) \\
&= 2a_{k+1}^2\tau D(\lambda_{k+1}) - 2a_k^2\tau D(\lambda_k) \qquad (50) \\
&\quad - 2(a_{k+1}^2 - a_k^2)\tau D(\lambda^\star) \\
&= 2a_k^2\tau\left(D(\lambda^\star) - D(\lambda_k)\right) \qquad (51) \\
&\quad + 2a_{k+1}^2\tau\left(D(\lambda_{k+1}) - D(\lambda^\star)\right)
\end{aligned}$$

where we have used the relation $a_k^2 = a_{k+1}^2 - a_{k+1}$ to traverse from (50) to (51). □

References

[1] J. M. Mendel and C. S. Currus, *Maximum-likelihood deconvolution: a journey into model-based signal processing*. Springer-Verlag, 1990.

[2] M. O'Brien, A. Sinclair, and S. Kramer, "Recovery of a sparse spike time series by ℓ_1 norm deconvolution," *IEEE Transactions on Signal Processing*, vol. 42, pp. 3353–3365, 1994.

[3] T. Olofsson and E. Wennerstrom, "Sparse deconvolution of b-scan images," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 54, pp. 1634–1641, 2007.

[4] T. Olofsson, "Semi-sparse deconvolution robust to uncertainties in the impulse responses," *Ultrasonics*, vol. 37, pp. 423–432, 1999.

[5] T. Olofsson and T. Stepinski, "Maximum a posteriori deconvolution of sparse ultrasonic signals using genetic optimization," *Ultrasonics*, vol. 37, no. 6, pp. 423 – 432, 1999.

[6] N. Galatsanos, A. Katsaggelos, R. Chin, and A. Hillery, "Least squares restoration of multichannel images," *IEEE Transactions on Signal Processing*, vol. 39, pp. 2222–2236, 1991.

[7] W. S. Ellis, S. J. Eisenberg, D. M. Auslander, M. W. Dae, A. Zakhor, and M. D. Lesh, "Deconvolution: A novel signal processing approach for determining activation time from fractionated electrograms and detecting infarcted tissue," *Circulation*, vol. 94, pp. 2633–2640, 1996.

[8] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica. D.*, vol. 60, pp. 259–268, 1992.

[9] J. J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bulletin de la S. M. F.*, vol. 93, pp. 273–299, 1965.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[11] R. T. Rockafellar and R. Wets, *Variational Analysis*, vol. 317 of *A Series of Comprehensive Studies in Mathematics*. Berlin: Springer, 2 ed., 2004.

[12] R. Glowinski and A. Marrocco *Inf. Rech. Oper.*, vol. R-2, pp. 41–76, 1975.

[13] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers amp; Mathematics with Applications*, vol. 2, no. 1, pp. 17 – 40, 1976.

[14] R. Glowinski and P. L. Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*. Philadephia, PA: Society for Industrial and Applied Mathematics, 1989.

[15] J. Eckstein and D. P. Bertsekas, "On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[16] T. Goldstein and S. Osher, "The split bregman method for ℓ_1 regularized problems," *UCLA CAM Report 08-29*, 2008.

[17] T. Goldstein, X. Bresson, and S. Osher, "Geometric applications of the split bregman method: Segmentation and surface reconstruction," *J. Sci. Comput.*, vol. 45, pp. 272–293, October 2010.

[18] P. Tseng, "Applications of splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM J. Control Optim.*, vol. 29, pp. 119–138, January 1991.

[19] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $o(1/k^2)$," *Soviet Math. Dokl.*, vol. 27, pp. 372–376, 1983.

[20] 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*, 2010.

[21] R. J. Bruck, "On the weak convergence of an ergodic iteration for the solution of variational inequalities for monotone operators in hilbert space," *J. Math. Anal. Appl.*, vol. 61, pp. 159–164, 1977.

[22] G. B. Passty, "Ergodic convergence to a zero of the sum of monotone operators in hilbert space," *J. Math. Anal. Appl.*, vol. 72, pp. 383–390, 1979.

[23] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[24] A. S. Nemirovsky and D. B. Yudin, *Problem complexity and method efficiency in optimization*. New York, NY: Wiley, 1983.

[25] Y. Nesterov, *Introductory Lectures on Convex Optimization*. New York, NY: Kluwer Academic Press, 2004.

[26] O. Güler, "On the convergence of the proximal point algorithm for convex minimization," *SIAM J. Control Optim.*, vol. 29, pp. 403–419, February 1991.

[27] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal of Imaging Sciences*, vol. 2, pp. 183–202, 2009.

[28] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, pp. 127–152, 2005. 10.1007/s10107-004-0552-5.

[29] S. Becker, J. Bobin, and E. Candes, "Nesta: A fast and accurate first-order method for sparse recovery," *Technical Report of Stanford University*, Apr 2009.

[30] S. Ji and J. Ye, "An accelerated gradient method for trace norm minimization," in *Proc. of the 26th Annual International Conference on Machine Learning*, ICML '09, (New York, NY, USA), pp. 457–464, ACM, 2009.

[31] D. Goldfarb, S. Ma, and K. Scheinberg, "Fast alternating linearization methods for minimizing the sum of two convex functions," *UCLA CAM technical report, 10-02*, 2010.

[32] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *UCLA CAM technical report, 12-52*, 2012.

[33] B. He and X. Yuan, "On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers," *Optimization Online*, Jan. 2012.

[34] M. Hong and Z.-Q. Luo, "On the Linear Convergence of the Alternating Direction Method of Multipliers," *ArXiv e-prints*, Mar. 2013.

[35] B. O'Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes." http://www-stat.stanford.edu/~candes/publications.html, 2012. Manuscript.

[36] R. T. Rockafellar, *Convex Analysis (Princeton Landmarks in Mathematics and Physics)*. Princeton University Press, Dec. 1996.

[37] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Convergence*, vol. 40, no. 1, pp. 1–49, 2010.

[38] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ: Princeton University Press, 1970.

[39] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society B*, vol. 67, pp. 301–320, 2005.

[40] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[41] M. Segal, K. Dahlquist, and B. Conklin, “Regression approach for microarray data analysis.,” *J. Computnl. Biol.*, vol. 10, pp. 961–980, 2002.

[42] A. M. Aibinu, M. J. E. Salami, A. A. Shafie, and A. R. Najeeb, “MRI reconstruction using discrete fourier transform: A tutorial,” *World Academy of Science, Engineering and Technology*, vol. 42, 2008.

[43] Z. P. Liang and P. C. Lauterbur, *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. Wiley-IEEE Press, Oct. 1999.

[44] E. J. Candes, J. Romberg, and T.Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 489 – 509, 2006.

[45] E. J. Candes and J. Romberg, “Signal recovery from random projections,” *Proc. of SPIE Computational Imaging III*, vol. 5674, pp. 76 – 86, 2005.

[46] M. Lustig, D. Donoho, and J. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging,” *Magnetic Resonance in Medicine*, vol. 58, pp. 1182–1195, 2007.

[47] S. S. Goh, A. Ron, and Z. Shen, *Mathematics and computation in imaging science and information processing*. World Scientific, 2007.

[48] T. F. Chan and C. K. Wong, “Total variation blind deconvolution,” *Image Processing, IEEE Transactions on*, vol. 7, pp. 370–375, Mar. 1998.

[49] T. Goldstein and S. Setzer, “High-order methods for basis pursuit,” *UCLA CAM technical report, 10-41*, 2010.

[50] L. Shepp and B. F. Logan, “Fourier reconstruction of a head section,” *IEEE Transactions on Nuclear Science*, vol. 21, no. 3, pp. 21–43, 1974.