An Acceleration of the Douglas Rachford Splitting Algorithm

February 1, 2014

Contents

Abstract

The bullets are from Tom's outline, now incorporated within the main document.

1 Introduction

- Why do we need splitting methods
- Common problem in imaging require large numberd of unknown

- "Big Data" applications have large number of unknowns
- We need methods that can handle sophisitcated problems, yet have "cheap" steps

This manuscript considers the problem

Problem 1. *minimize* $F(\lambda) = H(\lambda) + G(\lambda)$,

for convex H and G, where $\lambda \in \mathbb{R}^N$. In this manuscript, we will focus on solving problem 1 with the additional given:

Assumption 1. H and G have a Lipschitz continuous gradients, with Lipschitz constants $L(\nabla H) = \sigma_H$ and $L(\nabla G) = \sigma_G$.

With this additional assumption, we can prove a stronger convergent result for an accelerated Douglas Rachford splitting algorithm. Before we introduce the algorithm, we introduce a few important properties and definitions.

1.1 Terminology and Notation

- Define notation for norms, etc..
- Define strongly convex, convex conjugate, Lipschitz constant, and whatever else we use

If a function f has a Lipschitz continuous gradient with Lipschitz constant σ, then

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\sigma}{2}||y - x||^2. \tag{1}$$

1.2 Splitting Methods

1.2.1 Douglas Rachford Splitting

- History and references of inventors
- Introduce different forms of the method, and show their equivalence (e.g. repeat stuff from Ernie's paper)
- Mention that the method is also equivalent to ADMM
- State ADMM algortihm and discuss equivalence

Consider the following algorithm for finding a solution $\lambda^\star = \arg\min_\lambda F(\lambda)$, called Douglas-Rachford Splitting (DRS). In order to minimize $F(\lambda) = H(\lambda) + G(\lambda)$, we consider the problem of solving

$$\frac{d\lambda}{dt} + \nabla H(\lambda) + \nabla G(\lambda) = 0. \tag{2}$$

Then, iterating

$$\frac{\lambda_{k+1/2} - \lambda_k}{\tau} + \nabla H(\lambda_{k+1/2}) + \nabla G(\lambda_k) = 0 \tag{3}$$

$$\frac{\lambda_{k+1} - \lambda_k}{\tau} + \nabla H(\lambda_{k+1/2}) + \nabla G(\lambda_{k+1}) = 0 \tag{4}$$

to steady state produces a solution to $\nabla H(\lambda) + \nabla G(\lambda)$. Rearranging these iterates, we arrive at

$$\lambda_{k+1/2} + \tau \nabla H(\lambda_{k+1/2}) = \lambda_k - \tau \nabla G(\lambda_k) \tag{5}$$

$$\lambda_{k+1} + \tau \nabla G(\lambda_{k+1}) = \lambda_k - \tau \nabla H(\lambda_{k+1/2}). \tag{6}$$

The equation (5) can be written as

$$\lambda_{k+1/2} = J_{\tau \nabla H}(\lambda_k - \tau \nabla G(\lambda_k)), \tag{7}$$

and equation (6) is equivalent to

$$\lambda_{k+1} = J_{\tau \nabla G}(\lambda_k - \tau \nabla H(\lambda_{k+1/2})). \tag{8}$$

The steps in equations (7) and (8) form the core subroutine of the Douglas Rachford algorithms, which is given in algorithm 1.

Algorithm 1 `DRSstep`(λ^-)

Require: λ^-

1: $\lambda^{\frac{1}{2}} = J_{\tau \nabla H}(\lambda^- - \tau \nabla G(\lambda^-)) = \arg\min_{\lambda} \tau H(\lambda) + \frac{1}{2}||\lambda - \lambda^- + \tau \nabla G(\lambda^-)||^2$

2: $\lambda^+ = J_{\tau \nabla G}(\lambda^{\frac{1}{2}} + \tau \nabla G(\lambda^-)) = \arg\min_{\lambda} \tau G(\lambda) + \frac{1}{2}||\lambda - \lambda^{\frac{1}{2}} - \tau \nabla G(\lambda^-)||^2$

3: **return** λ^+

Douglas Rachford splitting is then prescribed by algorithm 2:

Algorithm 2 DRS

Require: $\lambda_0 \in R^N$

1: **for** $k = 0, 1, 2, \ldots$ **do**

2: $\quad \lambda_{k+1} =$ `DRSstep`(λ_k)

3: **end for**

1.3 Optimality Conditions and Residuals

The optimality conditions for algorithm 1 are:

$$\lambda^- - \lambda^{\frac{1}{2}} = \tau \left(\nabla G(\lambda^-) + \nabla H(\lambda^{\frac{1}{2}}) \right) \tag{9}$$

$$\lambda^+ - \lambda^{\frac{1}{2}} = \tau \left(\nabla G(\lambda^-) - \nabla G(\lambda^+) \right) \tag{10}$$

The residual for algorithm 2 is

$$r_k = ||\nabla G(\lambda_k) + \nabla H(\lambda_k)||. \tag{11}$$

2 Optimal Splitting Methods

Prior Work: State methods and give references.

2.1 Nesterov Acceleration

- Give some history: Gradient methods are very effective for simple problems, but they converge slow, and Nesterov invented a way to make them fast
- State gradient descent method. This method has $O(1/k)$ complexity.
- State Nesterov method, this has $O(1/k^2)$ complexity

2.2 Guler's Fast Proximal Point

2.3 FISTA

2.4 Goldfarb's Fast ADMM

2.5 Tom's Fast ADMM

2.6 FDRS

State that there's no known fast DRS. We propose a new fast DRS. the approach will follow Tom's approach for ADMM, but by working on DRS, we will prove stronger convergence results.

3 Preliminary Results

For the arguments presented here, we will leverage algorithm 1, and use the corresponding notation. We will also need the following alternate expression, which follows from equations (5) and (8):

$$\begin{aligned} \lambda^+ &= J_{\tau\nabla G}(\lambda^- - \tau\nabla H(\lambda^{\frac{1}{2}})) \\ &= \arg\min_{\lambda} \tau G(\lambda) + \frac{1}{2}||\lambda - \lambda^- + \tau\nabla H(\lambda^{\frac{1}{2}})||^2. \end{aligned} \tag{12}$$

From equation (12), we can derive an alternate (and equivalent) optimality condition for the second step of algorithm 1:

$$\lambda^- - \lambda^+ = \tau(\nabla G(\lambda^+) + \nabla H(\lambda^{\frac{1}{2}})) \tag{13}$$

Tom says: In this first result [this coming lemma], don't assume that the energy is differentiable. This lemma will hold if we either (a) use a line search to guarantee inequalities, or (b) have Lipschitz constants for the gradients.

Lemma 1. *Suppose that* $\tau^3 \leq \frac{1}{\sigma_H \sigma_G^2}$. *Then for any* $\gamma \in R^N$,

$$F(\gamma) - F(\lambda^+) \geq \tau^{-1}\langle \gamma - \lambda^-, \lambda^- - \lambda^+ \rangle + \frac{1}{2\tau}||\lambda - \lambda^+||^2$$

Proof. First, from equation (9), we have that

$$||\lambda^+ - \lambda^{\frac{1}{2}}|| \leq \tau\sigma_G||\lambda^- - \lambda^+||. \tag{14}$$

We proceed by using the convexity of H and equation (1).

$$\begin{aligned}
H(\gamma) - H(\lambda^+) &\geq H(\lambda^{\frac{1}{2}}) + \langle \nabla H(\lambda^{\frac{1}{2}}), \gamma - \lambda^{\frac{1}{2}} \rangle \\
&\quad - \left(H(\lambda^{\frac{1}{2}}) + \langle \nabla H(\lambda^{\frac{1}{2}}), \lambda^+ - \lambda^{\frac{1}{2}} \rangle + \frac{\sigma_H}{2}||\lambda^+ - \lambda^{\frac{1}{2}}||^2 \right) \\
&= \langle \nabla H(\lambda^{\frac{1}{2}}), \gamma - \lambda^+ \rangle - \frac{\sigma_H}{2}||\lambda^+ - \lambda^{\frac{1}{2}}||^2 \\
&\geq \langle \nabla H(\lambda^{\frac{1}{2}}), \gamma - \lambda^+ \rangle - \frac{\sigma_H \tau^2 \sigma_G^2}{2}||\lambda^- - \lambda^+||^2 \\
&\geq \langle \nabla H(\lambda^{\frac{1}{2}}), \gamma - \lambda^+ \rangle - \frac{1}{2\tau}||\lambda^- - \lambda^+||^2
\end{aligned}$$

By the convexity of G, we have

$$\begin{aligned}
G(\gamma) - G(\lambda^+) &\geq G(\lambda^+) + \langle \gamma - \lambda^+, \nabla G(\lambda^+) \rangle - G(\lambda^+) \\
&= \langle \gamma - \lambda^+, \nabla G(\lambda^+) \rangle.
\end{aligned}$$

By adding the estimates together, and using equation (13), we obtain

$$\begin{aligned}
F(\gamma) - F(\lambda^+) &\geq \langle \gamma - \lambda^+, \nabla H(\lambda^{\frac{1}{2}}) + \nabla G(\lambda^+) \rangle - \frac{1}{2\tau}||\lambda^- - \lambda^+||^2 \\
&= \frac{1}{\tau}\langle \gamma - \lambda^+, \lambda^- - \lambda^+ \rangle - \frac{1}{2\tau}||\lambda^- - \lambda^+||^2 \\
&= \frac{1}{\tau}\langle \gamma - \lambda^- + \lambda^- - \lambda^+, \lambda^- - \lambda^+ \rangle - \frac{1}{2\tau}||\lambda^- - \lambda^+||^2 \\
&= \frac{1}{\tau}\langle \gamma - \lambda, \lambda^- - \lambda^+ \rangle + \frac{1}{2\tau}||\lambda^- - \lambda^+||^2.
\end{aligned}$$

□

4 Global Convergence Bounds for Unaccelerated DRS

In this section we show that for DRS, we achieve the convergence $F(\lambda_k) - F(\lambda^\star) \leq O(1/k)$.

Theorem 1. *Consider FDRS described by algorithm 2. Suppose H and G satisfy assumption 1, and that $\tau^3 \leq \frac{1}{\sigma_H \sigma_G^2}$. Then for $k > 1$ the sequence $\{\lambda_k\}$ satisfies*

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

where $\lambda^\star$ minimizes F.

Proof. We begin exploiting Lemma 1 with $\gamma = \lambda^\star$ and $\lambda^- = \lambda_k$. Then, $\lambda^+ = \lambda_{k+1}$.

$$\begin{aligned} 2\tau(F(\lambda^\star) - F(\lambda_{k+1})) &\geq 2\langle \lambda^\star - \lambda_k, \lambda_k - \lambda_{k+1}\rangle + ||\lambda_k - \lambda_{k+1}||^2 \\ &= ||\lambda^\star - \lambda_{k+1}||^2 - ||\lambda^\star - \lambda_k||^2. \end{aligned}$$

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

$$2\tau\left((n-1)F(\lambda^\star) - \sum_{k=1}^{n-1} F(\lambda_{k+1})\right) \geq ||\lambda^\star - \lambda_n||^2 - ||\lambda^\star - \lambda_1||^2. \tag{15}$$

Using Lemma 1 again with $\lambda^- = \gamma = \lambda_k$ gives

$$2\tau\left(F(\lambda_k) - F(\lambda_{k+1})\right) \geq ||\lambda_k - \lambda_{k+1}||^2. \tag{16}$$

Multiplying equation (16) by $k-1$, and summing over $k = 1, 2, \ldots, n-1$ produces

$$2\tau\left(\sum_{k=1}^{n-1}(k-1)\left[F(\lambda_k) - F(\lambda_{k+1})\right]\right) \geq \sum_{k=1}^{n-1}(k-1)||\lambda_k - \lambda_{k+1}||^2. \tag{17}$$

The telescoping sum on the left of equation (17) gives

$$2\tau\left(\sum_{k=1}^{n-1} F(\lambda_{k+1}) - (n-1)F(\lambda_n)\right) \geq \sum_{k=1}^{n-1}(k-1)||\lambda_k - \lambda_{k+1}||^2. \tag{18}$$

Adding equation (15) and equation (18) yields the bound

$$2\tau(n-1)\left[F(\lambda^\star) - F(\lambda_n)\right] \geq ||\lambda^\star - \lambda_n||^2 - ||\lambda^\star - \lambda_1||^2 + \sum_{k=1}^{n-1}(k-1)||\lambda_k - \lambda_{k+1}||^2 \tag{19}$$

$$\geq -||\lambda^\star - \lambda_1||^2. \tag{20}$$

It follows that

$$F(\lambda_n) - F(\lambda^\star) \leq \frac{||\lambda^\star - \lambda_1||^2}{2\tau(n-1)}.$$

□

5 Fast Douglas Rachford Splitting

In this section, we consider a Nesterov like acceleration of the DRS algorithm, which we term Fast Douglas-Rachford Splitting (FDRS).

The optimality conditions for the above algorithm are:

Algorithm 3 FDRS

Require: $\alpha_0 = 1, \lambda_{-1} = \hat{\lambda}_0 \in R^N$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: $\quad \lambda_k = \texttt{DRSstep}(\hat{\lambda}_k)$
3: $\quad \alpha_{k+1} = \frac{1+\sqrt{1+4\alpha_k^2}}{2}$
4: $\quad \hat{\lambda}_{k+1} = \lambda_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\lambda_k - \lambda_{k-1})$
5: **end for**

$$\hat{\lambda}_k - \lambda_{k+1/2} = \tau \left(\nabla G(\hat{\lambda}_k) + \nabla H(\lambda_{k+1/2}) \right) \tag{21}$$

$$\lambda_k - \lambda_{k+1/2} = \tau \left(\nabla G(\hat{\lambda}_k) - \nabla G(\lambda_k) \right) \tag{22}$$

$$\hat{\lambda}_k - \lambda_k = \tau \left(\nabla G(\lambda_k) + \nabla H(\lambda_{k+1/2}) \right) \text{(alternate)}. \tag{23}$$

Define

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

Lemma 2. *Let λ_k, $\hat{\lambda}_k$ and α_k be defined as in algorithm 3. Then*

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

Proof. First note that from the definition of $\hat{\lambda}_{k+1}$ we have the identity

$$\alpha_{k+1}(\hat{\lambda}_{k+1} - \lambda_k) = (\alpha_k - 1)(\lambda_k - \lambda_{k-1}). \tag{24}$$

Using this observation in the definition of s_k produces

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

□

Lemmas 1 and 2 combine for the following estimate.

Lemma 3. *Suppose that H and G satisfy Assumption 1 and that G satisfies Assumption 2. The iterates generated by algorithm 3 without restart and the sequence $\{s_k\}$ obey the following relation:*

$$||s_{k+1}||^2 - ||s_k||^2 \leq 2\alpha_k^2 \tau \left(F(\lambda^\star) - F(\lambda_k) \right) - 2\alpha_{k+1}^2 \tau \left(F(\lambda_{k+1}) - F(\lambda^\star) \right).$$

Proof. To begin we apply Lemma 1 with $\gamma = \lambda_k$ and $\lambda^- = \hat{\lambda}_{k+1}$, which gives $\lambda^+ = \lambda_{k+1}$. We then have the following bound:

$$F(\lambda_k) - F(\lambda_{k+1}) \geq \frac{1}{\tau}\langle \hat{\lambda}_{k+1} - \lambda_k, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle + \frac{1}{2\tau}||\lambda_{k+1} - \hat{\lambda}_{k+1}||^2. \tag{25}$$

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

$$F(\lambda^\star) - F(\lambda_{k+1}) \geq \frac{1}{\tau}\langle \hat{\lambda}_{k+1} - \lambda^\star, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle + \frac{1}{2\tau}||\lambda_{k+1} - \hat{\lambda}_{k+1}||^2. \tag{26}$$

Next, from Lemma 2, we have

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

Rearranging this yields

$$\begin{aligned}
||s_{k+1}||^2 - ||s_k||^2 &= 2\alpha_{k+1}\langle s_k, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle + \alpha_{k+1}^2||\lambda_{k+1} - \hat{\lambda}_{k+1}||^2 \\
&= 2\alpha_{k+1}\langle \alpha_k\lambda_k - (\alpha_k - 1)\lambda_{k-1} - \lambda^\star, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle \\
&\quad + \alpha_{k+1}^2||\lambda_{k+1} - \hat{\lambda}_{k+1}||^2 \\
&= 2\alpha_{k+1}\langle \alpha_{k+1}\hat{\lambda}_{k+1} + (1 - \alpha_{k+1})\lambda_k - \lambda^\star, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle \\
&\quad + \alpha_{k+1}^2||\lambda_{k+1} - \hat{\lambda}_{k+1}||^2 \quad \text{from identity equation (24)} \\
&= 2\alpha_{k+1}\langle (\alpha_{k+1} - 1)(\hat{\lambda}_{k+1} - \lambda_k) + \hat{\lambda}_{k+1} - \lambda^\star, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle \\
&\quad + \alpha_{k+1}^2||\lambda_{k+1} - \hat{\lambda}_{k+1}||^2 \\
&= 2\alpha_{k+1}(\alpha_{k+1} - 1)\langle \hat{\lambda}_{k+1} - \lambda_k, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle \\
&\quad + 2\alpha_{k+1}\langle \hat{\lambda}_{k+1} - \lambda^\star, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle + \alpha_{k+1}^2||\lambda_{k+1} - \hat{\lambda}_{k+1}||^2 \\
&= 2\alpha_{k+1}(\alpha_{k+1} - 1)\left(\langle \hat{\lambda}_{k+1} - \lambda_k, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle + \frac{1}{2}||\lambda_{k+1} - \hat{\lambda}_{k+1}||^2\right) \\
&\quad + 2\alpha_{k+1}\left(\langle \hat{\lambda}_{k+1} - \lambda^\star, \lambda_{k+1} - \hat{\lambda}_{k+1}\rangle + \frac{1}{2}||\lambda_{k+1} - \hat{\lambda}_{k+1}||^2\right).
\end{aligned}$$

Observe that $\alpha_{k+1} = \alpha_{k+1}^2 - \alpha_k^2$. Using this, and the estimates equation (25) and equation (26) with the above equality yields

$$
\begin{aligned}
||s_{k+1}||^2 - ||s_k||^2 &\leq 2\alpha_{k+1}\tau(\alpha_{k+1}-1)(F(\lambda_k) - F(\lambda_{k+1})) \\
&\quad + 2\alpha_{k+1}\tau(F(\lambda^\star) - F(\lambda_{k+1})) \\
&= 2\alpha_{k+1}(\alpha_{k+1}-1)\tau F(\lambda_k) + 2\alpha_{k+1}\tau F(\lambda^\star) \\
&\quad - 2\alpha_{k+1}^2\tau F(\lambda_{k+1}) \\
&= 2\alpha_k^2\tau F(\lambda_k) + 2\left(\alpha_{k+1}^2 - \alpha_k^2\right)\tau F(\lambda^\star) \\
&\quad - 2\alpha_{k+1}^2\tau F(\lambda_{k+1}) \\
&= 2\alpha_k^2\tau\left[F(\lambda_k) - F(\lambda^\star)\right] - 2\alpha_{k+1}^2\tau\left[F(\lambda_{k+1}) - F(\lambda^\star)\right].
\end{aligned}
$$

□

Theorem 2. *Suppose that H and G satisfy Assumption 1 and that G satisfies Assumption 2. The iterates λ_k generated by algorithm 3 without restart satisfy*

$$
F(\lambda_k) - F(\lambda^\star) \leq \frac{2\tau||\hat{\lambda}_0 - \lambda^\star||^2}{(k+2)^2}.
$$

Proof. Lemma 3 can be rearranged as

$$
||s_{k+1}||^2 + 2\alpha_{k+1}^2\tau\left[F(\lambda_{k+1}) - F(\lambda^\star)\right] \leq ||s_k||^2 + 2\alpha_k^2\tau\left[F(\lambda_k) - F(\lambda^\star)\right].
$$

Recalling that $\alpha_0 = 1$, induction then implies that

$$
||s_{k+1}||^2 + 2\alpha_{k+1}^2\tau\left[F(\lambda_{k+1}) - F(\lambda^\star)\right] \leq ||s_0||^2 + 2\tau\left[F(\lambda_0) - F(\lambda^\star)\right], \tag{27}
$$

Then applying equation (26) with $k = 0$ gives

$$
F(\lambda^\star) - F(\lambda_0) \geq \frac{1}{\tau}\langle\hat{\lambda}_0 - \lambda^\star, \lambda_0 - \hat{\lambda}_0\rangle + \frac{1}{2\tau}||\lambda_0 - \hat{\lambda}_0||^2 = \frac{1}{2\tau}\left(||\lambda_0 - \lambda^\star||^2 - ||\hat{\lambda}_0 - \lambda^\star||^2\right). \tag{28}
$$

Applying estimates equation (27) and equation (28) to Lemma 3 we have

$$
\begin{aligned}
2\alpha_{k+1}^2\tau(F(\lambda_{k+1}) - F(\lambda^\star)) &\leq ||s_k||^2 - ||s_{k+1}||^2 + 2\alpha_k^2\tau\left[F(\lambda_k) - F(\lambda^\star)\right] \\
&\leq ||s_k||^2 + 2\alpha_k^2\tau\left[F(\lambda_k) - F(\lambda^\star)\right] \\
&\leq ||s_0||^2 + 2\tau\left[F(\lambda_0) - F(\lambda^\star)\right] \\
&= ||\lambda_0 - \lambda^\star||^2 + 2\tau\left[F(\lambda_0) - F(\lambda^\star)\right] \\
&\leq ||\lambda_0 - \lambda^\star||^2 + ||\hat{\lambda}_0 - \lambda^\star||^2 - ||\lambda_0 - \lambda^\star||^2 \\
&= ||\hat{\lambda}_0 - \lambda^\star||^2.
\end{aligned}
$$

It then follows that

$$
F(\lambda_k) - F(\lambda^\star) \leq \frac{||\hat{\lambda}_0 - \lambda^\star||^2}{2\alpha_k^2\tau}.
$$

Using the observation that $\alpha_k > \alpha_{k-1} + \frac{1}{2} > 1 + \frac{k}{2}$ gives

$$F(\lambda_k) - F(\lambda^\star) \leq \frac{2||\hat{\lambda}_0 - \lambda^\star||^2}{(k+2)^2\tau}.$$

□

6 Fast Douglas Rachford Splitting for Non-Differential Problems

We consider a variant of algorithm 3 in which we impose a restart condition, which reverts the algorithm to the traditional Douglas Rachford Splitting method in certain cases, which ensures convergence. The restart rule relies on a combined residual, which bounds the residual for problem 1:

$$\begin{aligned} r_k &= ||\nabla G(\lambda_k) + \nabla H(\lambda_k)|| \\ &\leq ||\nabla G(\lambda_k) + \nabla H(\lambda_{k+1/2})|| \\ &\quad + ||\nabla H(\lambda_k) - \nabla H(\lambda_{k+1/2})|| =: c_k. \end{aligned} \tag{29}$$

The algorithm is then given by algorithm 4. Notice that Lines 1-2 produce $\lambda_{k+1} = \texttt{DRSstep}(\hat{\lambda}_k)$, as in algorithm 3.

Algorithm 4 FDRS with Restart

Require: $\alpha_0 = 1, \lambda_{-1} = \hat{\lambda}_0 \in R^N$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: $\quad \lambda_{k+1/2} = J_{\tau\nabla H}(\hat{\lambda}_k - \tau\nabla G(\hat{\lambda})) = \arg\min_\lambda \tau H(\lambda) + \frac{1}{2}||\lambda - \hat{\lambda}_k + \tau\nabla G(\hat{\lambda}_k)||^2$
3: $\quad \lambda_{k+1} = J_{\tau\nabla G}(\lambda_{k+1/2} + \tau\nabla G(\hat{\lambda})) = \arg\min_\lambda \tau G(\lambda) + \frac{1}{2}||\lambda - \lambda_{k+1/2} - \tau\nabla G(\hat{\lambda}_k)||^2$
4: $\quad c_k = ||\nabla G(\lambda_{k+1}) + \nabla H(\lambda_{k+1/2})|| + ||\nabla H(\lambda_{k+1}) - \nabla H(\lambda_{k+1/2})||$
5: $\quad$ **if** $c_k < \eta c_{k-1}$ **then**
6: $\quad\quad \alpha_{k+1} = \frac{1+\sqrt{1+4\alpha_k^2}}{2}$
7: $\quad\quad \hat{\lambda}_{k+1} = \lambda_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\lambda_k - \lambda_{k-1})$
8: $\quad$ **else**
9: $\quad\quad \alpha_{k+1} = 1,\ \lambda_{k+1/2} = \lambda_{k-1}$
10: $\quad\quad c_k \leftarrow \eta^{-1} c_{k-1}$
11: $\quad$ **end if**
12: **end for**

Lemma 4. *The iterates $\{\lambda_k, \lambda_{k+1/2}\}$ produced in algorithm 4 satisfy*

$$\begin{aligned} &||\nabla G(\lambda_{k+1}) + \nabla H(\lambda_{k+3/2})|| + ||\nabla H(\lambda_{k+1}) - \nabla H(\lambda_{k+3/2})|| \\ &\leq ||\nabla G(\lambda_k) + \nabla H(\lambda_{k+1/2})|| + ||\nabla H(\lambda_k) - \nabla H(\lambda_{k+1/2})||. \end{aligned} \tag{30}$$

Proof. Blah blah blah cite Yuan and He. (But they do it for ADMM....) □

Theorem 3. *For convex H and G, algorithm 4 converges:*

$$\lim_{k\to\infty} c_k = 0. \tag{31}$$

Proof. We begin with some terminology. Each iteration of algorithm 4 is of one of three types:

1. A restart iteration occurs when the inequality in Step 5 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 6-7) of algorithm 4 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 6-7 of the algorithm are invoked and $\alpha_k > 1$.

Suppose that a restart occurs at iteration k. Then the value λ_k are returned to their values at iteration $k-1$, which has combined residual c_{k-1}. We also set $\alpha_{k+1} = 1$, making iteration $k+1$ an unaccelerated iteration. By Lemma 4 the combined residual is non-increasing on this iteration and so $c_{k+1} \leq c_{k-1}$. Note that on accelerated steps, the combined residual decreases by at least a factor of η. It follows that the combined residual satisfies

$$c_k \leq c_0 \eta^{\hat{k}}, \tag{32}$$

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

Clearly, if the number of accelerated iterations is infinite, then we have $c_k \to 0$ as $k \to \infty$. 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 DRS (algorithm 2) for which convergence is known cite[]. □

While algorithm 4 enables us to extend accelerated DRS to non-differential problems, our theoretical results are weaker in this case because we cannot guarantee a convergence rate as we did under assumption 1. 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].

7 Backtracking

State the line search method. I (TOM) will do some experiments to figure out the best way to do this, but try to structure the Lemma in the preliminaries section so the Lipschitz gradient part is separate from the main inequality Lemma.

8 Relation to Other Work

- Global Convergence rates for DRS have also been proved by Binsheng He (I'll dig this paper up for you), but without any acceleration.
- This is similar to the ADMM method, however our results are a stronger because of the line search - we can gaurantee convergence for non-differentiable problems (i.e. the duals of weakly convex problems that would normally be solved with ADMM)

9 Numerical Results

- Do some simple quadratic programming examples
- We need to come up with a few more examples of things that DRS is good for. There's tons of these, but we have to pick a few that are easy to code.