Proximal Mappings Involving Almost Structured Matrices

İlker Bayram

Abstract—**We consider a minimization problem where the cost function consists of the sum of a quadratic data fidelity term and a penalty term. The quadratic involves a matrix H that can be embedded into a larger matrix $\tilde{H}$ where multiplication with the inverse of $I + \alpha \tilde{H}^T \tilde{H}$ can be efficiently performed. We discuss how to take advantage of this property when the Douglas-Rachford algorithm is utilized.**

I. Introduction

Consider a minimization problem containing a quadratic data fidelity term and a penalty term as,

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|y - H\,x\|_2^2 + P(x), \tag{P1}$$

where H is an $m \times n$ matrix, $y \in \mathbb{R}^m$ is a given data vector, $\|\cdot\|_2$ denotes the ℓ_2 norm on $\mathbb{R}^m$ and $P(\cdot)$ is a convex penalty function (or regularizer). When the proximals (see Defn. 1) of the data fidelity term and the penalty term are easy to evaluate, this problem can be efficiently solved using the Douglas-Rachford algorithm [5]. The proximal of the quadratic data term requires multiplication with $(I + \alpha\, H^T H)^{-1}$, where '$I$' denotes the identity matrix and 'α' is a positive parameter (see Ex. 1 below). If H is a structured matrix like a circulant matrix, or if it is sparse etc., then this multiplication can be performed in a computationally efficient manner. In this letter, we consider the case where H does not enjoy such a property but by adding rows and/or columns it can be completed to a larger $\tilde{H}$ for which $(I + \alpha\, \tilde{H}^T \tilde{H})^{-1}$ is computationally feasible to compute.

A primary case of interest is when H is associated with linear convolution. In fact, if H has a Toeplitz structure determined by a filter h, then multiplication with H or $I + \alpha\, H^T H$ can be realized using fast Fourier transforms (fft) via zero padding (see e.g. Sec. 8.7.2 of [11]). Therefore, handling such matrices is easy for forward-backward type algorithms which only employ multiplication with H or H^T [6]. However, the Douglas-Rachford algorithm requires multiplication with $(I + \alpha\, H^T H)^{-1}$ which requires convolution with an inverse filter and this may not be easily realized with fft's (see e.g. [10, 1]) because the inverse filter might be infinite impulse response. Nevertheless, if h is short, $I + \alpha\, H^T H$ will be banded and can be inverted efficiently (see e.g. the discussion on banded systems in [8]). In a context similar to this letter's, this point has been noted and exploited in [14] to accelerate the convergence of a majorization-minimization algorithm. Unfortunately, when h has many non-zero coefficients, it is not computationally feasible to invert $I + \alpha\, H^T H$. Nevertheless, H can be completed to a larger circulant $\tilde{H}$ [16], for which multiplication with $(I + \alpha\, \tilde{H}^T \tilde{H})^{-1}$ can be realized efficiently via fft's.

Another example is when y is obtained by irregularly sampling a noisy observation of a signal obtained from a dictionary [13], where the dictionary atoms are taken from a frame with a synthesis operator that can be implemented efficiently [4]. In that case, the matrix H whose columns hold the dictionary atoms is obtained by irregularly sampling the rows and columns of the frame synthesis operator $\tilde{H}$ and it might be computationally more desirable to multiply with $(I + \alpha\, \tilde{H}^T \tilde{H})^{-1}$ instead of $(I + \alpha\, H^T H)^{-1}$.

Related Work

The problem introduced above has been considered in the context of handling boundary conditions in image deconvolution [10, 1, 15, 12]. Specifically, in [15, 12], the authors assume that pixels outside a specified boundary do not affect the observed image y. This in turn leads to a quadratic data term as in (P1) where H can be completed to a circulant matrix by appending rows, i.e., $\tilde{H} = \begin{bmatrix} H^T & G^T \end{bmatrix}^T$ for some G. Noting that $H^T H = \tilde{H}^T \tilde{H} - G^T G$, it is then possible to resort to the matrix inversion lemma to write (see [15, 12] for the slightly different context/problem)

$$(I + \alpha\, H^T H)^{-1} = A^{-1} + \alpha\, A^{-1} G^T B^{-1} G\, A^{-1}, \tag{1}$$

where $A = (I + \alpha\, \tilde{H}^T \tilde{H})$ and $B = (I - \alpha\, G\, A^{-1} G^T)$. Multiplication with B^{-1} can be efficiently implemented if G has a small number of rows since then B will have a small size. However, multiplication with B^{-1} may be as hard to realize as $(I + \alpha\, H^T H)^{-1}$ when G has many rows. An alternative is presented in [10, 1] where the authors consider a data fidelity term of the form $\frac{1}{2}\|y - T\,\tilde{H}\,x\|_2^2$, where T is a wide diagonal matrix such that $T\,\tilde{H} = H$. Note that inverting $\left(I + (T\,\tilde{H})^T\,(T\,\tilde{H})\right)$ is not computationally feasible either. Therefore, this transformation is not suitable for a direct implementation of the Douglas-Rachford algorithm. In order to avoid computing this inverse, the authors employ variable splitting and use the alternating direction method of multipliers (ADMM) algorithm [7, 3]. However, as will be shown in the sequel, it is also possible to directly employ the Douglas-Rachford algorithm without variable splitting.

Proposed Approach

We propose to consider a modified problem equivalent to (P1) which can be written as the sum of two terms with

İ. Bayram is with the Dept. of Electronics and Telecommunications Eng., Istanbul Technical University, Istanbul, Turkey. E-mail : ibayram@itu.edu.tr.

This research is supported by TÜBİTAK (Project No : 113E511).

simple proximal mappings, thus facilitating the realization of Douglas-Rachford iterations. The equivalent problem allows to append both rows and columns to H to reach $\tilde{H}$ (see Section V for the description of a problem which requires that). Thus, the considered problem is slightly more general than those in [10, 1, 15, 12], which are interested in appending rows to H.

Outline

We start with a brief discusson on the Douglas-Rachford algorithm in Section II. The discussion of the equivalent problem and the resulting Douglas-Rachford iterations are discussed in Section III. In Section IV we consider two special cases where we need to append only columns or only rows to H. A batch of experiments with varying problem sizes is presented in Section V. Section VI is the conclusion.

II. The Douglas-Rachford Algorithm

Before we state the algorithm, let us recall the definition of a proximity operator.

Definition 1. The *proximity operator (proximal mapping)* of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ is denoted by $J_{\alpha f}$ and is defined for $\alpha > 0$ as

$$J_{\alpha f}(x') = \arg\min_x \frac{1}{2}\|x' - x\|_2^2 + \alpha\, f(x). \tag{2}$$

We refer to [5] for a table of proximal mappings that are relevant from a signal processing viewpoint. An example of interest is the following.

Example 1. For $f(x) = \dfrac{1}{2}\|y - H\,x\|_2^2$, the proximity operator is given as,

$$J_{\alpha f}(x') = \arg\min_x \frac{1}{2}\|x' - x\|_2^2 + \frac{\alpha}{2}\|y - H\,x\|_2^2 \tag{3a}$$

$$= (I + \alpha\, H^T H)^{-1}(x' + \alpha\, H^T y). \tag{3b}$$

For a minimization problem of the form

$$\min_x f(x) + g(x), \tag{4}$$

where both f and g are convex, the Douglas-Rachford iterations consist of

$$x^{n+1} = \Big(\lambda\, I + (1-\lambda)\,\Big(2J_{\alpha g} - I\Big)\,\Big(2J_{\alpha f} - I\Big)\Big)\; x^n, \tag{5}$$

for $\lambda \in (0,1)$. Regardless of the value of α, the sequence x^n converges to a point x^* such that $J_{\alpha f}(x^*)$ minimizes $f + g$. We refer to [9, 7, 2] for a study of convergence. For (P1), pseudocode for the Douglas-Rachford algorithm is presented in Algorithm 1. The 'repeat' loop in this algorithm yields an approximate x^* and the last step is the final proximal operation $J_{\alpha f}(x^*)$ mentioned above.

III. Appending Rows and Columns to H

Suppose H is $m \times n$ and it can be embedded into $\tilde{H}$ which is $m' \times n'$, with $m' \geq m$, $n' \geq n$. In general, the rows and columns of $\tilde{H}$ can be sampled in any order but for simplicity, we assume that H and $\tilde{H}$ are related as,

$$\tilde{H} = \begin{bmatrix} H & H_1 \\ H_2 & H_3 \end{bmatrix}, \tag{6}$$

Algorithm 1 Douglas-Rachford Algorithm for (P1)

Initialize $x \in \mathbb{R}^n$
repeat
  $\tilde{x} \leftarrow 2(I + \alpha\, H^T H)^{-1}\,(x + \alpha\, H^T y) - x$
  $x \leftarrow \lambda\, x + (1-\lambda)\big(2\, J_{\alpha\, P}(\tilde{x}) - \tilde{x}\big)$
until some convergence criterion is met
$x^* \leftarrow (I + \alpha\, H^T H)^{-1}\, x$

for some matrices H_1, H_2, H_3. If we directly apply the Douglas-Rachford algoritm to problem (P1), we cannot employ the operator $(I + \alpha\, \tilde{H}^T \tilde{H})^{-1}$ which is assumed to be easily realizable. Our idea is to first define an equivalent problem and employ the Douglas-Rachford algorithm on this modified problem.

A. A Modified Problem

In the setting described above, consider the problem,

$$\min_{x,z,t} \frac{1}{2}\left\| t - \tilde{H}\, \begin{bmatrix} x \\ z \end{bmatrix} \right\|_2^2 + P(x),$$

$$\text{s.t.} \begin{cases} t_k = y_k, & \text{for } 1 \leq k \leq m, \\ z = 0. \end{cases} \tag{P2}$$

where $(x, z, t) \in \mathbb{R}^n \times \mathbb{R}^{n'-n} \times \mathbb{R}^{m'}$ and t_k, y_k denote the k^{th} entry of t, y respectively. The seemingly vacuous constraint $z = 0$ is in fact necessary for our stated purpose of avoiding $(I + \alpha\, H^T H)^{-1}$ and using $(I + \alpha\, \tilde{H}^T \tilde{H})^{-1}$ instead. The Douglas-Rachford iterations will need to keep track of the variables z, which will assume non-zero values throughout the iterations and converge to zero only in the limit.

The two problems (P1) and (P2) are equivalent in the following sense.

Proposition 1. $x^* \in \mathbb{R}^n$ is a solution of (P1) if and only if $(x^*, z^*, t^*) \in \mathbb{R}^{n+(n'-n)+m'}$ is a solution of (P2) for some $z^* \in \mathbb{R}^{n'-n}$, $t^* \in \mathbb{R}^{m'}$.

Proof. Due to the constraints on (P2), we can restate it as,

$$\min_{x,\tilde{t}} \left\{ C_2(x, \tilde{t}) = \frac{1}{2}\left\| \begin{bmatrix} y \\ \tilde{t} \end{bmatrix} - \tilde{H}\, \begin{bmatrix} x \\ 0 \end{bmatrix} \right\|_2^2 + P(x) \right\}, \tag{7}$$

where $(x, \tilde{t}) \in \mathbb{R}^n \times \mathbb{R}^{(m'-m)}$. Using (6), we can write,

$$C_2(x, \tilde{t}) = \underbrace{\frac{1}{2}\,\|y - H\,x\|_2^2 + P(x)}_{C_1(x)} + \frac{1}{2}\,\|\tilde{t} - H_2\,x\|_2^2. \tag{8}$$

Therefore $C_2(x, \tilde{t}) \geq C_1(x)$ for all $(x, \tilde{t})$.

Suppose now that x^* solves (P1). This is equivalent to $C_1(x^*) \leq C_1(x)$ for all x. Now let $\tilde{t}^* = H_2\, x^*$. We have,

$$C_2(x^*, \tilde{t}^*) = C_1(x^*) \leq C_1(x) \leq C_2(x, \tilde{t}) \tag{9}$$

for any $(x, \tilde{t})$. Thus $\left(x^*, 0, \begin{bmatrix} y \\ \tilde{t}^* \end{bmatrix}\right)$ solves (P2).

For the converse, suppose $\left(x^*, 0, \begin{bmatrix} y \\ \tilde{t}^* \end{bmatrix}\right)$ solves (P2), but x^* does not minimize (P1). Then there exists some x' such that $C_1(x') < C_1(x^*)$. But then for $\tilde{t}' = H_2\, x'$, we would have

$$C_2(x', \tilde{t}') = C_1(x') < C_1(x^*) \leq C_2(x^*, \tilde{t}^*), \tag{10}$$

contradicting the assumption that $\left(x^*, 0, \begin{bmatrix} y \\ \tilde{t}^* \end{bmatrix}\right)$ solves (P2). □

B. Douglas-Rachford Iterations for (P2)

For the Douglas-Rachford iterations, we split (P2) as,

$$\min_{(x,z,t)} \underbrace{\frac{1}{2} \left\| t - \tilde{H} \begin{bmatrix} x \\ z \end{bmatrix} \right\|_2^2}_{f(x,t,z)} + \underbrace{P(x) + i_C(t,z)}_{g(x,t,z)} \tag{11}$$

where $i_C(t,z)$ is a characteristic function enforcing the constraints, defined as

$$i_C(t,z) = \begin{cases} 0, & \text{if } \begin{cases} t_k = y_k, & \text{for } 1 \le k \le m, \\ z = 0, \end{cases} \\ \infty, & \text{otherwise.} \end{cases}$$

Given f and g as in (11), we need expressions for their proximals. First, note that

$$J_{\alpha f}(x', z', t') = \arg \min_{(x,z,t)} \frac{1}{2}\|x - x'\|_2^2 + \frac{1}{2}\|z - z'\|_2^2 + \frac{1}{2}\|t - t'\|_2^2 + \frac{\alpha}{2} \left\| t - \tilde{H} \begin{bmatrix} x \\ z \end{bmatrix} \right\|_2^2. \tag{12}$$

For $(\hat{x}, \hat{z}, \hat{t}) = J_{\alpha f}(x', z', t')$, we find, by the optimality conditions that

$$\begin{bmatrix} (1+\alpha)\, I & -\alpha\, \tilde{H} \\ -\alpha\, \tilde{H}^T & I + \alpha\, \tilde{H}^T \tilde{H} \end{bmatrix} \begin{bmatrix} \hat{t} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} t' \\ u' \end{bmatrix}, \tag{13}$$

where

$$\hat{u} = \begin{bmatrix} \hat{x} \\ \hat{z} \end{bmatrix}, \quad u' = \begin{bmatrix} x' \\ z' \end{bmatrix}. \tag{14}$$

Solving (13), we find,

$$\hat{u} = \left(I + \frac{\alpha}{1+\alpha} \tilde{H}^T \tilde{H} \right)^{-1} \left(u' + \frac{\alpha}{1+\alpha} \tilde{H}^T t' \right) \tag{15a}$$

$$\hat{t} = \frac{1}{1+\alpha} \left(t' + \alpha\, \tilde{H}\, \hat{u} \right). \tag{15b}$$

We also note that if $(\hat{x}, \hat{z}, \hat{t}) = J_{\alpha g}(x', z', t')$, then

$$\hat{x} = J_{\alpha P}(x'), \tag{16a}$$

$$\hat{z} = 0, \tag{16b}$$

$$\hat{t}_k = \begin{cases} y_k & \text{if } 1 \le k \le m, \\ t'_k & \text{if } m+1 \le k \le m'. \end{cases} \tag{16c}$$

Using these expressions for the proximals, we can write down the Douglas-Rachford iterations. After some rearrangements, we obtain the iterations in Algorithm 2. Notice that Algorithm 2 employs only the inverse of $(I + \beta\, \tilde{H}^T \tilde{H})$.

IV. Special Cases

In order to obtain $\tilde{H}$ we may need to append only rows or only columns to H. In those cases, Algorithm 2 can be simplified.

Algorithm 2 Douglas-Rachford Alg. for (11), Solving (P2)

1: Initialize $x \in \mathbb{R}^n$, $t \in \mathbb{R}^{m'}$, $z \in \mathbb{R}^{(n'-n)}$, $\beta \leftarrow \dfrac{\alpha}{1+\alpha}$

2: **repeat**

3: $\quad \begin{bmatrix} \tilde{x} \\ \tilde{z} \end{bmatrix} \leftarrow \left(I + \beta\, \tilde{H}^T \tilde{H} \right)^{-1} \left(\begin{bmatrix} x \\ z \end{bmatrix} + \beta\, \tilde{H}^T t \right)$

4: $\quad \tilde{t} \leftarrow \dfrac{1-\alpha}{1+\alpha}\, t + 2\beta\, \tilde{H} \begin{bmatrix} \tilde{x} \\ \tilde{z} \end{bmatrix}$

5: $\quad x \leftarrow x + 2(1-\lambda) \left(J_{\alpha P}(2\tilde{x} - x) - \tilde{x} \right)$

6: $\quad z \leftarrow z - 2(1-\lambda)\, \tilde{z}$

7: $\quad t_k \leftarrow \begin{cases} \lambda\, t_k + (1-\lambda)\,(2y_k - \tilde{t}_k), & \text{for } 1 \le k \le m, \\ \lambda\, t_k + (1-\lambda)\, \tilde{t}_k, & \text{for } m+1 \le k \le m' \end{cases}$

8: **until** some convergence criterion is met

9: $\begin{bmatrix} x^* \\ z^* \end{bmatrix} \leftarrow \left(I + \beta\, \tilde{H}^T \tilde{H} \right)^{-1} \left(\begin{bmatrix} x \\ z \end{bmatrix} + \beta\, \tilde{H}^T t \right)$

A. Appending Columns to H

Suppose H is $m \times n$ and $\tilde{H}$ is $m \times n'$ with $n' > n$. Assume for simplicity that H can be embedded into $\tilde{H}$ as

$$\tilde{H} = \begin{bmatrix} H & H_1 \end{bmatrix} \tag{17}$$

for an $m \times (n' - n)$ matrix H_1. Consider the problem

$$\min_{x,z} \frac{1}{2} \left\| y - \tilde{H} \begin{bmatrix} x \\ z \end{bmatrix} \right\|_2^2 + P(x), \ \text{s.t. } z = 0, \tag{P3}$$

where $(x, z) \in \mathbb{R}^n \times \mathbb{R}^{n'-n}$. The problem (P3) is equivalent to (P1) in the following sense.

Proposition 2. $x^* \in \mathbb{R}^n$ is a solution of (P1) if and only if $(x^*, 0) \in \mathbb{R}^n \times \mathbb{R}^{(n'-n)}$ is a solution of (P3). □

Now let

$$f(x,z) = \frac{1}{2} \left\| y - \tilde{H} \begin{bmatrix} x \\ z \end{bmatrix} \right\|_2^2, \tag{18a}$$

$$g(x,z) = \begin{cases} P(x), & \text{if } z = 0, \\ \infty, & \text{if } z \ne 0. \end{cases} \tag{18b}$$

Then, for $(\hat{x}, \hat{z}) = J_{\alpha f}(x', z')$, we have,

$$\begin{bmatrix} \hat{x} \\ \hat{z} \end{bmatrix} = (I + \alpha\, \tilde{H}^T \tilde{H})^{-1} \left(\begin{bmatrix} x' \\ z' \end{bmatrix} + \alpha\, \tilde{H}^T y \right). \tag{19}$$

Also, if $(\hat{x}, \hat{z}) = J_{\alpha g}(x', z')$, then, $\hat{x} = J_{\alpha P}(x')$, and $\hat{z} = 0$.

The resulting Douglas-Rachford iterations are given in Algorithm 3. Note that these iterations are much simpler than those in Algorithm 2 since we do not need to replace the constant y with constrained variables.

B. Appending Rows to H

Suppose now that H is $m \times n$, $\tilde{H}$ is $m' \times n$ with $m' > m$ and the two matrices are related as, $\tilde{H} = \begin{bmatrix} H^T & H_1^T \end{bmatrix}^T$, for an $(m'-m) \times n$ matrix H_1. An algorithm for this special case can be obtained by removing the variable z from the discussion in Section III. Consequently the problem we consider is

$$\min_{x \in R^n,\, t \in \mathbb{R}^{m'}} \frac{1}{2} \left\| t - \tilde{H}\, x \right\|_2^2 + P(x), \quad \text{s.t. } t_k = y_k, \ \text{for } 1 \le k \le m. \tag{P4}$$

Algorithm 3 Douglas-Rachford Algorithm for (P3)

Initialize $x \in \mathbb{R}^n$, $z \in \mathbb{R}^{(n'-n)}$.
repeat
$$\begin{bmatrix}\tilde{x}\\ \tilde{z}\end{bmatrix} \leftarrow 2\left(I + \alpha\,\tilde{H}^T\,\tilde{H}\right)^{-1}\left(\begin{bmatrix}x\\ z\end{bmatrix} + \alpha\,\tilde{H}^T\,y\right) - \begin{bmatrix}x\\ z\end{bmatrix}$$
$$x \leftarrow \lambda\,x + (1-\lambda)\left(2J_{\alpha\,P}(\tilde{x}) - \tilde{x}\right)$$
$$z \leftarrow \lambda\,z - (1-\lambda)\,\tilde{z}$$
until some convergence criterion is met
$$\begin{bmatrix}x^*\\ z^*\end{bmatrix} \leftarrow \left(I + \alpha\,\tilde{H}^T\,\tilde{H}\right)^{-1}\left(\begin{bmatrix}x\\ z\end{bmatrix} + \alpha\,\tilde{H}^T\,y\right)$$

An algorithm for solving this problem can be obtained by removing z and $\tilde{z}$ from Algorithm 2. The resulting pseudocode is given in Algorithm 4.

Algorithm 4 Douglas-Rachford Algorithm for (P4)

Initialize $x \in \mathbb{R}^n$, $t \in \mathbb{R}^{m'}$, set $\beta \leftarrow \alpha/(1+\alpha)$
repeat
$$\tilde{x} \leftarrow \left(I + \beta\,\tilde{H}^T\,\tilde{H}\right)^{-1}\left(x + \beta\,\tilde{H}^T\,t\right)$$
$$\tilde{t} \leftarrow \frac{1-\alpha}{1+\alpha}\,t + 2\beta\,\tilde{H}\,\tilde{x}$$
$$x \leftarrow x + 2(1-\lambda)\left(J_{\alpha\,P}(2\tilde{x} - x) - \tilde{x}\right)$$
$$t_k \leftarrow \begin{cases}\lambda\,t_k + (1-\lambda)\,(2y_k - \tilde{t}_k), & \text{for } 1 \le k \le m,\\ \lambda\,t_k + (1-\lambda)\,\tilde{t}_k, & \text{for } m+1 \le k \le m'\end{cases}$$
until some convergence criterion is met
$$x^* \leftarrow \left(I + \beta\,\tilde{H}^T\,\tilde{H}\right)^{-1}\left(x + \beta\,\tilde{H}^T\,t\right)$$

V. Experiments

In order to assess the convergence speed of the proposed modification, we performed tests involving problems of varying sizes. Our purpose here is to not to provide theoretical limits but to demonstrate that the proposed modification can be effective especially for large problems. For the experiments, we reach $\tilde{H}$ by appending rows *and* columns to H. Therefore this problem falls out of the immediate scope of the discussion in [10, 1, 15, 12]. Matlab code for these experiments can be found at "`http://web.itu.edu.tr/ibayram/Structured/`".

We took H as $N \times N$ Toeplitz matrices where $N = k \cdot 10^4$ for $k = 1, 2, \ldots, 10$. The matrices H are associated with a causal filter h (so that H is lower-triangular) of length $K_h = 2000$. The filter h is produced randomly by sampling a length-K_h random vector with independent entries which are uniformly distributed on $[0, 1]$ and multiplying the resulting vector with an exponentially decaying mask. Note that H can be embedded in a circulant $(N + K_h - 1) \times (N + K_h - 1)$ matrix $\tilde{H}$. We took x to be a random sparse vector (95% zero, non-zeros are obtained by sampling from a standard Gaussian distribution) and produced y as, $y = Hx + u$, where u denotes Gaussian noise with variance σ^2, chosen such that SNR = 10 dB. We took $P(x) = \tau\|x\|_1$ for $\tau = 3\sigma$, so that the proximal of P, namely $J_{\alpha P}$, is a soft-threshold with threshold equal to $\alpha\,\tau$. We also set the parameters of the algorithm as $\alpha = 0.1$, $\lambda = 0.1$, which lead to a fair convergence

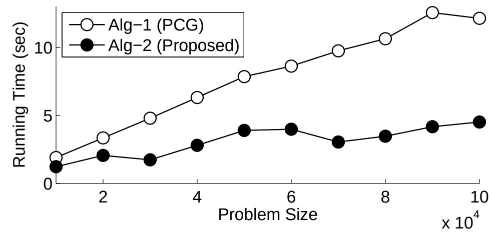

Fig. 1. Running times for the regular Douglas-Rachford algorithm (Algorithm 1) for 20 iterations and Algorithm 2 for 400 iterations.

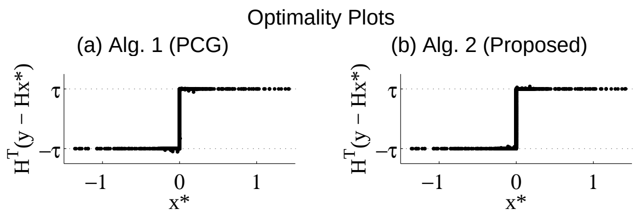

Fig. 2. Optimality plots for a problem of size 10^5, showing $H^T(y - Hx^*)$ vs. x^*, where x^* is the solution obtained by (a) Algorithm 1, (b) Algorithm 2.

behavior. In this setting, we ran the regular Douglas-Rachford iterations (Algorithm 1) and Algorithm 2. Due to the size of the matrices, the operation $(I + \alpha H^T H)^{-1}$ in Algorithm 1 is realized using preconditioned conjugate gradients (PCG) [17]. For preconditioning, we used an $N \times N$ circulant M associated with the filter $\delta(n) + \alpha\,a_h(n)$, where $a_h(n)$ denotes the autocorrelation function of the filter h (see e.g. [16] or Lecture 40 of [17]). Note that M can be inverted efficiently using fft's. For Algorithm 2, we realized multiplication with $(I + \alpha\tilde{H}^T\,\tilde{H})^{-1}$ also using fft's (observe however that this matrix is larger in size compared to M). In our experiments, we observed that Algorithm 2 requires more iterations to approximately converge to the optimal solution, due probably to the increased number of variables. For this reason, we ran Algorithm 1 for 20 iterations and Algorithm 2 for 400 iterations. For these choices, the algorithms approximately converged. We also made sure that 400 iterations Algorithm 2 returns a lower cost than 20 iterations of Algorithm 1. The running times for a regular PC are shown in Fig. 1.

In order to check whether the algorithms converged, we also viewed the optimality conditions. For our problem, x^* is a solution if and only if $H^T(y - Hx^*) \in \tau\,\mathrm{sign}(x^*)$. For H of size $10^5 \times 10^5$, the optimality plots showing x^* vs. $H^T(y - H\,x^*)$ for the two algorithms are provided in Fig. 2. We observe that the solutions obtained by both algorithms approximately satisfy the optimality conditions.

To conclude, these experiments demonstrate that the proposed modified problem and the resulting iterations can lead to a considerable reduction in the running time.

VI. Conclusion

We considered an implementation detail pertaining to algorithms involving proximals of quadratics. Efficient realization of this proximal can be important especially for large problems, which are of growing interest in signal processing. Although we specifically considered the Douglas-Rachford algorithm, the proposed approach can be useful for other algorithms that employ proximals such as ADMM, forward-backward or majorization-minimization algorithms.

REFERENCES

[1] M. S. C. Almeida and M. A. T. Figueiredo. Deconvolving images with unknown boundaries using the alternating direction method of multipliers. *IEEE Trans. Image Proc.*, 22(8):3074–3086, May 2013.

[2] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011.

[3] 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*, 3(1):1–122, 2011.

[4] O. Christensen. *An Introduction to Frames and Riesz Bases*. Birkhäuser, 2003.

[5] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer, New York, 2011.

[6] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM J. Multiscale Model. Simul.*, 4(4):1168–1200, November 2005.

[7] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(3):293–318, 1992.

[8] G. H. Golub and C. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

[9] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16(6):964–979, 1979.

[10] A. Matakos, S. Ramani, and J. A. Fessler. Accelerated edge-preserving image restoration without boundary artifacts. *IEEE Trans. Image Proc.*, 22(5):2019–2029, May 2013.

[11] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Pearson, third edition, 2010.

[12] S. J. Reeves. Fast image restoration without boundary artifacts. *IEEE Trans. Image Proc.*, 14(10):1448 – 1453, October 2005.

[13] R. Rubinstein, A.M. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proc. IEEE*, 98(6):1045–1057, June 2010.

[14] I. W. Selesnick, H. L. Graber, D. S. Pfeil, and R. L. Barbour. Simultaneous low-pass filtering and total variation denoising. *IEEE Trans. Signal Processing*, 62(5):1109–1124, March 2014.

[15] M. Sorel. Removing boundary artifacts for real-time iterated shrinkage deconvolution. *IEEE Trans. Image Proc.*, 21(4):2329 – 2334, April 2012.

[16] G. Strang. A proposal for Toeplitz matrix calculations. *Studies in Applied Mathematics*, 74(2):171–176, April 1986.

[17] L. N. Trefethen and D. Bau III. *Numerical Linear Algebra*. SIAM, 1997.