Adaptive Penalty Parameter Selection for ADMM Algorithms

Brendt Wohlberg

EDICS: OPT-CVXR, OPT-SOPT

***Abstract*—Appropriate selection of the penalty parameter is crucial to obtaining good performance from the Alternating Direction Method of Multipliers (ADMM). While analytic results for optimal selection of this parameter are very limited, there is a heuristic method that appears to be relatively successful in a number of different problems. The contribution of this paper is to demonstrate that their is a potentially serious flaw in the standard definition of the residuals used in this heuristic approach, and to propose a method that rectifies this flaw, as well as a additional modifications that further improves the efficacy of this method.**

***Index Terms*—ADMM, penalty parameter, sparse representation**

I. Introduction

The Alternating Direction Method of Multipliers (ADMM) has become a very popular approach to solving a broad variety of optimization problems in signal and image processing, prominent examples including Total Variation regularization and sparse representation problems [1], [2, Sec. 6], [3]. This method introduces an additional parameter, the *penalty parameter*, on which the rate of convergence is strongly dependent, but for which there are no analytic results to guide selection other than for a very specific set of problems [4], [5]. There is, however, a heuristic method for automatically adapting the penalty parameter [6] that appears to becoming quite popular [7], [8], [9], [10], [11], [12]. The present paper introduces some improvements to this approach that are illustrated with computational experiments involving sparse representation problems.

II. ADMM

The notation and exposition in this section follows that of the influential tutorial by Boyd et al. [2]. The *Lagrangian* for the constrained problem

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) \;\text{ such that }\; A\mathbf{x} = \mathbf{b} \;, \tag{1}$$

is

$$L(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + \mathbf{y}^T (A\mathbf{x} - \mathbf{b}) \;. \tag{2}$$

The *method of multipliers* solves this problem via *dual ascent*

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{y}^{(k)}) \tag{3}$$

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \rho(A\mathbf{x}^{(k+1)} - \mathbf{b}) \;, \tag{4}$$

Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA. Email: brendt@lanl.gov, Tel: +1 505 667 6886

This research was supported by the U.S. Department of Energy through the LANL/LDRD Program.

where L_ρ is the *augmented Lagrangian*

$$L_\rho(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + \mathbf{y}^T (A\mathbf{x} - \mathbf{b}) + \frac{\rho}{2} \left\| A\mathbf{x} - \mathbf{b} \right\|_2^2 \tag{5}$$

with *penalty parameter* ρ.

ADMM can be viewed (although there are limitations to this interpretation [13]) as a variant of this method applied to the problem

$$\arg\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \;\text{ such that }\; A\mathbf{x} + B\mathbf{z} = \mathbf{c} \;, \tag{6}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^p$, and the Lagrangian and augmented Lagrangian are, respectively,

$$L(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^T (A\mathbf{x} + B\mathbf{z} - \mathbf{c}) \tag{7}$$

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = L(\mathbf{x}, \mathbf{z}, \mathbf{y}) + \frac{\rho}{2} \left\| A\mathbf{x} + B\mathbf{z} - \mathbf{c} \right\|_2^2 \;. \tag{8}$$

Instead of jointly solving for $\mathbf{x}$ and $\mathbf{z}$, ADMM alternates the $\mathbf{x}$ and $\mathbf{z}$ updates (thus the *alternating direction*)

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{z}^{(k)}, \mathbf{y}^{(k)}) \tag{9}$$

$$\mathbf{z}^{(k+1)} = \arg\min_{\mathbf{z}} L_\rho(\mathbf{x}^{(k+1)}, \mathbf{z}, \mathbf{y}^{(k)}) \tag{10}$$

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \rho(A\mathbf{x}^{(k+1)} + B\mathbf{z}^{(k+1)} - \mathbf{c}) \;. \tag{11}$$

It is often more convenient to work with the *scaled form* of ADMM, which is obtained by the change of variable to the *scaled dual variable* $\mathbf{u} = \rho^{-1}\mathbf{y}$. Defining the residual

$$\mathbf{r} = A\mathbf{x} + B\mathbf{z} - \mathbf{c} \tag{12}$$

and replacing $\mathbf{y}$ with $\mathbf{u}$ we have

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{u}) = f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2} \left\| \mathbf{r} + \mathbf{u} \right\|_2^2 - \frac{\rho}{2} \left\| \mathbf{u} \right\|_2^2 \;. \tag{13}$$

Since the minimisers of $L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{u})$ with respect to $\mathbf{x}$ and $\mathbf{z}$ do not depend on the final $\frac{\rho}{2} \left\| \mathbf{u} \right\|_2^2$ term, the iterations can be written as

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2} \left\| A\mathbf{x} + B\mathbf{z}^{(k)} - \mathbf{c} + \mathbf{u}^{(k)} \right\|_2^2 \tag{14}$$

$$\mathbf{z}^{(k+1)} = \arg\min_{\mathbf{y}} g(\mathbf{z}) + \frac{\rho}{2} \left\| A\mathbf{x}^{(k+1)} + B\mathbf{z} - \mathbf{c} + \mathbf{u}^{(k)} \right\|_2^2 \tag{15}$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + A\mathbf{x}^{(k+1)} + B\mathbf{z}^{(k+1)} - \mathbf{c} \;. \tag{16}$$

III. ADMM Residuals

The primal feasibility condition for Eq. (6) is

$$A\mathbf{x}^* + B\mathbf{z}^* - \mathbf{c} = 0 \;, \tag{17}$$

and the dual feasibility conditions are

$$0 \in \partial L(\cdot, \mathbf{z}^*, \mathbf{y}^*) \;\Rightarrow\; 0 \in \partial f(\mathbf{x}^*) + A^T \mathbf{y}^* \tag{18}$$

$$0 \in \partial L(\mathbf{x}^*, \cdot, \mathbf{y}^*) \;\Rightarrow\; 0 \in \partial g(\mathbf{z}^*) + B^T \mathbf{y}^* \;, \tag{19}$$

where ∂ denotes the subdifferential operator [14, Ch. D].

Since $\mathbf{z}^{(k+1)}$ minimises $L_\rho(\mathbf{x}^{(k+1)}, \mathbf{z}, \mathbf{y}^{(k)})$ (see Eq. (10)), we have

$$\begin{aligned}
0 &\in [\partial L_\rho(\mathbf{x}^{(k+1)}, \cdot, \mathbf{y}^{(k)})](\mathbf{z}^{(k+1)}) && (20)\\
&= \partial g(\mathbf{z}^{(k+1)}) + B^T \mathbf{y}^{(k)} \\
&\qquad + \rho B^T (A\mathbf{x}^{(k+1)} + B\mathbf{z}^{(k+1)} - \mathbf{c}) && (21)\\
&= \partial g(\mathbf{z}^{(k+1)}) + B^T \mathbf{y}^{(k)} + \rho B^T \mathbf{r}^{(k+1)} && (22)\\
&= \partial g(\mathbf{z}^{(k+1)}) + B^T (\mathbf{y}^{(k)} + \rho \mathbf{r}^{(k+1)}) && (23)\\
&= \partial g(\mathbf{z}^{(k+1)}) + B^T \mathbf{y}^{(k+1)} \;, && (24)
\end{aligned}$$

so that iterates $\mathbf{z}^{(k+1)}$ and $\mathbf{y}^{(k+1)}$ always satisfy dual feasibility condition Eq. (19), leaving Eq. (17) and Eq. (18) as the remaining optimality criteria to be satisfied. Similarly, since $\mathbf{x}^{(k+1)}$ minimises $L_\rho(\mathbf{x}, \mathbf{z}^{(k)}, \mathbf{y}^{(k)})$ (see Eq. (9)), we have

$$\begin{aligned}
0 &\in [\partial L_\rho(\cdot, \mathbf{z}^{(k)}, \mathbf{y}^{(k)})](\mathbf{x}^{(k+1)}) && (25)\\
&= \partial f(\mathbf{x}^{(k+1)}) + A^T \mathbf{y}^{(k)} \\
&\qquad + \rho A^T (A\mathbf{x}^{(k+1)} + B\mathbf{z}^{(k)} - \mathbf{c}) && (26)\\
&= \partial f(\mathbf{x}^{(k+1)}) + A^T \mathbf{y}^{(k)} + \rho A^T (A\mathbf{x}^{(k+1)} + B\mathbf{z}^{(k+1)} \\
&\qquad - \mathbf{c} + B\mathbf{z}^{(k)} - B\mathbf{z}^{(k+1)}) && (27)\\
&= \partial f(\mathbf{x}^{(k+1)}) + A^T \mathbf{y}^{(k)} \\
&\qquad + \rho A^T (\mathbf{r}^{(k+1)} + B\mathbf{z}^{(k)} - B\mathbf{z}^{(k+1)}) && (28)\\
&= \partial f(\mathbf{x}^{(k+1)}) + A^T (\mathbf{y}^{(k)} \\
&\qquad + \rho \mathbf{r}^{(k+1)}) + \rho A^T B(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)}) && (29)\\
&= \partial f(\mathbf{x}^{(k+1)}) + A^T \mathbf{y}^{(k+1)} + \rho A^T B(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)}) \;, && (30)
\end{aligned}$$

which can be written as

$$\rho A^T B(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)}) \in \partial f(\mathbf{x}^{(k+1)}) + A^T \mathbf{y}^{(k+1)} \;. \tag{31}$$

This suggests defining

$$\mathbf{s}^{(k+1)} = \rho A^T B(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)}) \tag{32}$$

as a residual for the dual feasibility condition Eq. (18). The corresponding residual for the primal feasibility condition is

$$\mathbf{r}^{(k+1)} = A\mathbf{x}^{(k+1)} + B\mathbf{z}^{(k+1)} - \mathbf{c} \;. \tag{33}$$

A. Stopping Criteria and Adaptive Penalty Parameter

These residuals play an important role in defining stopping criteria for the ADMM iterations, e.g. Boyd et al. [2, Sec. 3.3.1] recommend stopping criteria

$$\left\|\mathbf{r}^{(k)}\right\|_2 \leq \epsilon_{\text{pri}}^{(k)} \quad \text{and} \quad \left\|\mathbf{s}^{(k)}\right\|_2 \leq \epsilon_{\text{dua}}^{(k)} \tag{34}$$

where

$$\epsilon_{\text{pri}}^{(k)} = \sqrt{p}\epsilon_{\text{abs}} + \epsilon_{\text{rel}} \max\left\{\left\|A\mathbf{x}^{(k)}\right\|_2, \left\|B\mathbf{z}^{(k)}\right\|_2, \left\|\mathbf{c}\right\|_2\right\} \tag{35}$$

$$\epsilon_{\text{dua}}^{(k)} = \sqrt{n}\epsilon_{\text{abs}} + \epsilon_{\text{rel}} \left\|A^T\mathbf{y}^{(k)}\right\|_2 \;, \tag{36}$$

ϵ_{abs} and ϵ_{rel} are absolute and relative tolerances respectively, and n and p are the dimensionalities of $\mathbf{x}$ and $\mathbf{c}$ respectively (i.e. $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{c} \in \mathbb{R}^p$).

These residuals have also been used in a scheme for automatically adapting the penalty parameter [6] [2, Sec 3.4.1] that has has been found to be effective for a variety of problems [7], [8], [9], [10], [11], [12]. The update is

$$\rho^{(k+1)} = \begin{cases} \tau\rho^{(k)} & \text{if } \left\|\mathbf{r}^{(k)}\right\|_2 > \mu\left\|\mathbf{s}^{(k)}\right\|_2 \\ \tau^{-1}\rho^{(k)} & \text{if } \left\|\mathbf{s}^{(k)}\right\|_2 > \mu\left\|\mathbf{r}^{(k)}\right\|_2 \\ \rho^{(k)} & \text{otherwise} \end{cases} \;, \tag{37}$$

where τ and μ are constants, the usual values being $\tau = 2$ and $\mu = 10$ [6], [15], [2]. This scheme, which is based on varying ρ to balance the primal and dual residual norms, can be very effective as a method for selecting the penalty parameter, the correct choice of which plays a vital role in obtaining good convergence.

IV. ADMM Problem Scaling Properties

We now turn to consider the behaviour of ADMM under scaling of the optimization problem being addressed. Denote Eq. (6) as problem P and define $\tilde{P}$ as

$$\arg\min_{\mathbf{x},\mathbf{z}} \alpha f(\gamma\mathbf{x}) + \alpha g(\gamma\mathbf{z}) \;\text{ s.t. }\; \beta A\gamma\mathbf{x} + \beta B\gamma\mathbf{z} = \beta\mathbf{c} \;, \tag{38}$$

with Lagrangian

$$\begin{aligned}
\tilde{L}(\mathbf{x}, \mathbf{z}, \mathbf{y}) &= \alpha f(\gamma\mathbf{x}) + \alpha g(\gamma\mathbf{z}) \\
&\qquad + \mathbf{y}^T(\beta\gamma A\mathbf{x} + \beta\gamma B\mathbf{z} - \beta\mathbf{c}) \;.
\end{aligned} \tag{39}$$

In the standard form, this problem is

$$\arg\min_{\mathbf{x},\mathbf{z}} \tilde{f}(\mathbf{x}) + \tilde{g}(\mathbf{z}) \;\text{ such that }\; \tilde{A}\mathbf{x} + \tilde{B}\mathbf{z} = \tilde{\mathbf{c}} \tag{40}$$

with

$$\begin{aligned}
\tilde{f}(\mathbf{x}) = \alpha f(\gamma\mathbf{x}) \quad \tilde{g}(\mathbf{x}) = \alpha g(\gamma\mathbf{x}) \\
\tilde{A} = \beta\gamma A \quad \tilde{B} = \beta\gamma B \quad \tilde{\mathbf{c}} = \beta\mathbf{c} \;.
\end{aligned} \tag{41}$$

The primal feasibility condition for this problem is

$$\beta A\gamma\tilde{\mathbf{x}}^* + \beta B\gamma\tilde{\mathbf{z}}^* - \beta\tilde{\mathbf{c}} = 0 \;, \tag{42}$$

and the dual feasibility conditions are

$$0 \in \partial\tilde{L}(\cdot, \tilde{\mathbf{z}}^*, \tilde{\mathbf{y}}^*) \;\Rightarrow 0 \in \alpha\gamma[\partial f(\cdot)](\gamma\tilde{\mathbf{x}}^*) + \beta\gamma A^T\tilde{\mathbf{y}}^* = 0 \tag{43}$$

$$0 \in \partial\tilde{L}(\tilde{\mathbf{x}}^*, \cdot, \tilde{\mathbf{y}}^*) \;\Rightarrow 0 \in \alpha\gamma[\partial g(\cdot)](\gamma\tilde{\mathbf{z}}^*) + \beta\gamma B^T\tilde{\mathbf{y}}^* = 0 \;. \tag{44}$$

It is easily verified that if $\mathbf{x}^*$, $\mathbf{z}^*$, and $\mathbf{y}^*$ satisfy the optimality criteria Eq. (17), (18), and (19) for problem P, then

$$\tilde{\mathbf{x}}^* = \gamma^{-1}\mathbf{x}^* \qquad \tilde{\mathbf{z}}^* = \gamma^{-1}\mathbf{z}^* \qquad \tilde{\mathbf{y}}^* = \frac{\alpha}{\beta}\mathbf{y}^* \tag{45}$$

satisfy the primal and dual feasibility criteria for $\tilde{P}$. The augmented Lagrangian for $\tilde{P}$ is

$$\begin{aligned}
\tilde{L}_{\tilde{\rho}}(\mathbf{x}, \mathbf{z}, \mathbf{y}) &= \alpha f(\gamma\mathbf{x}) + \alpha g(\gamma\mathbf{z}) \\
&\qquad + \alpha\left(\frac{\beta}{\alpha}\mathbf{y}^T\right)(\gamma A\mathbf{x} + \gamma B\mathbf{z} - \mathbf{c}) \\
&\qquad + \alpha\left(\frac{\beta^2}{\alpha}\tilde{\rho}\right)\frac{1}{2}\left\|\gamma A\mathbf{x} + \gamma B\mathbf{z} - \mathbf{c}\right\|_2^2 \;,
\end{aligned} \tag{46}$$

so that setting $\tilde{\rho} = \alpha\rho/\beta^2$ gives

$$\tilde{L}_{\tilde{\rho}}(\mathbf{x}, \mathbf{z}, \mathbf{y}) = \alpha L_{\rho}\left(\gamma\mathbf{x}, \gamma\mathbf{z}, \frac{\beta}{\alpha}\mathbf{y}\right) \; . \tag{47}$$

The results $\mathbf{x}^{(k+1)}$, $\mathbf{z}^{(k+1)}$, and $\mathbf{y}^{(k+1)}$ for iteration k of the ADMM algorithm for P are given by Eq. (9), (10), and (11). We now consider the corresponding iterates for $\tilde{P}$, assuming that

$$\tilde{\mathbf{z}}^{(k)} = \gamma^{-1}\mathbf{z}^{(k)} \qquad \tilde{\mathbf{y}}^{(k)} = \frac{\alpha}{\beta}\mathbf{y}^{(k)} \; . \tag{48}$$

The $\mathbf{x}$ update is

$$\begin{aligned} \tilde{\mathbf{x}}^{(k+1)} &= \arg\min_{\mathbf{x}} \tilde{L}_{\tilde{\rho}}(\mathbf{x}, \tilde{\mathbf{z}}^{(k)}, \tilde{\mathbf{y}}^{(k)}) \\ &= \arg\min_{\mathbf{x}} \tilde{L}_{\tilde{\rho}}(\mathbf{x}, \gamma^{-1}\mathbf{z}^{(k)}, \frac{\alpha}{\beta}\mathbf{y}^{(k)}) \\ &= \arg\min_{\mathbf{x}} \alpha L_{\rho}(\gamma\mathbf{x}, \mathbf{z}^{(k)}, \mathbf{y}^{(k)}) \; . \end{aligned} \tag{49}$$

For convex f we have that if $\mathbf{x}^*$ minimises $f(\mathbf{x})$ then $\gamma^{-1}\mathbf{x}^*$ minimises $g(\mathbf{x}) = \alpha f(\gamma\mathbf{x})$, so

$$\tilde{\mathbf{x}}^{(k+1)} = \gamma^{-1}\mathbf{x}^{(k+1)} \; , \tag{50}$$

and similarly it can be shown that

$$\tilde{\mathbf{z}}^{(k+1)} = \gamma^{-1}\mathbf{z}^{(k+1)} \; . \tag{51}$$

For the $\mathbf{y}$ update we have

$$\begin{aligned} \tilde{\mathbf{y}}^{(k+1)} &= \tilde{\mathbf{y}}^{(k)} + \tilde{\rho}(\beta\gamma A\tilde{\mathbf{x}}^{(k+1)} + \beta\gamma B\tilde{\mathbf{z}}^{(k+1)} - \beta\mathbf{c}) \\ &= \frac{\alpha}{\beta}\left(\mathbf{y}^{(k)} + \rho(A\mathbf{x}^{(k+1)} + B\mathbf{z}^{(k+1)} - \mathbf{c})\right) \\ &= \frac{\alpha}{\beta}\mathbf{y}^{(k+1)} \; . \end{aligned} \tag{52}$$

A. Residual Scaling

Now, consider the primal and dual residuals for $\tilde{P}$

$$\begin{aligned} \tilde{\mathbf{r}}^{(k+1)} &= \tilde{A}\tilde{\mathbf{x}}^{(k+1)} + \tilde{B}\tilde{\mathbf{z}}^{(k+1)} - \tilde{\mathbf{c}} \\ &= \beta A\mathbf{x}^{(k+1)} + \beta B\mathbf{z}^{(k+1)} - \beta\mathbf{c} \\ &= \beta\mathbf{r}^{(k+1)} \end{aligned} \tag{53}$$

$$\begin{aligned} \tilde{\mathbf{s}}^{(k+1)} &= \tilde{\rho}\tilde{A}^T\tilde{B}(\tilde{\mathbf{z}}^{(k)} - \tilde{\mathbf{z}}^{(k+1)}) \\ &= \alpha\gamma\rho A^T B(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)}) \\ &= \alpha\gamma\mathbf{s}^{(k+1)} \; . \end{aligned} \tag{54}$$

The important observation here is that the primal and dual residuals do not share the same scaling factors. This scaling behaviour is highly undesirable since it has a potentially severe effect on the adaptive penalty scheme based on balancing the two residuals[1]. It is clear that if adapting ρ so that $\left\|\tilde{\mathbf{r}}^{(k+1)}\right\| / \left\|\tilde{\mathbf{s}}^{(k+1)}\right\| \approx 1$ functions correctly for some values of α, β, and γ, it can not be expected to do so for substantially different values, and there is no reason to expect that the most natural way of setting up a problem will always correspond to the choices of these variables for which the method does function properly.

The obvious solution is to normalise the residuals by appropriate quantities that will cancel the scaling with β and $\alpha\gamma$, making them invariant to problem scaling. The normalisations in the definitions of $\epsilon_{\mathrm{pri}}^{(k+1)}$ and $\epsilon_{\mathrm{dua}}^{(k+1)}$ (see Eq. (35) and (36)) are, in fact, suitable for this purpose, since

$$\begin{aligned} &\max\left\{\left\|\tilde{A}\tilde{\mathbf{x}}^{(k+1)}\right\|_2, \left\|\tilde{B}\tilde{\mathbf{z}}^{(k+1)}\right\|_2, \left\|\tilde{\mathbf{c}}\right\|_2\right\} \\ &= \max\left\{\left\|\beta\gamma A\gamma^{-1}\mathbf{x}^{(k+1)}\right\|_2, \left\|\beta\gamma B\gamma^{-1}\mathbf{z}^{(k+1)}\right\|_2, \left\|\beta\mathbf{c}\right\|_2\right\} \\ &= \beta\max\left\{\left\|A\mathbf{x}^{(k+1)}\right\|_2, \left\|B\mathbf{z}^{(k+1)}\right\|_2, \left\|\mathbf{c}\right\|_2\right\} \end{aligned} \tag{55}$$

and

$$\left\|\tilde{A}^T\tilde{\mathbf{y}}^{(k+1)}\right\|_2 = \left\|\beta\gamma A^T\frac{\alpha}{\beta}\mathbf{y}^{(k+1)}\right\|_2 = \alpha\gamma\left\|A^T\mathbf{y}^{(k+1)}\right\|_2. \tag{56}$$

If we define

$$\mathbf{r}^{(k+1)} = \frac{A\mathbf{x}^{(k+1)} + B\mathbf{z}^{(k+1)} - \mathbf{c}}{\max\left\{\left\|A\mathbf{x}^{(k+1)}\right\|_2, \left\|B\mathbf{z}^{(k+1)}\right\|_2, \left\|\mathbf{c}\right\|_2\right\}} \tag{57}$$

$$\begin{aligned} \mathbf{s}^{(k+1)} &= \frac{\rho A^T B(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)})}{\left\|A^T\mathbf{y}^{(k+1)}\right\|_2} \\ &= \frac{A^T B(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)})}{\left\|A^T\mathbf{u}^{(k+1)}\right\|_2} \end{aligned} \tag{58}$$

then $\tilde{\mathbf{r}}^{(k+1)} = \mathbf{r}^{(k+1)}$ and $\tilde{\mathbf{s}}^{(k+1)} = \mathbf{s}^{(k+1)}$. The convergence proof [6] of the standard adaptive scheme (i.e. Eq. (37) with the standard definitions of the residuals) depends only on bounds on the sequences $\rho^{(k)}$ and $\eta_k = \sqrt{(\rho^{(k+1)}/\rho^{(k)})^2 - 1}$, neither of which is affected by the change in the definition of the residuals, so the convergence results still hold under the modified definitions of the residuals. In order to maintain equivalence with the original stopping criteria, the correspondingly modified definitions of the stopping tolerances become

$$\epsilon_{\mathrm{pri}}^{(k+1)} = \frac{\sqrt{p}\epsilon_{\mathrm{abs}}}{\max\left\{\left\|A\mathbf{x}^{(k+1)}\right\|_2, \left\|B\mathbf{z}^{(k+1)}\right\|_2, \left\|\mathbf{c}\right\|_2\right\}} + \epsilon_{\mathrm{rel}} \tag{59}$$

$$\epsilon_{\mathrm{dua}}^{(k+1)} = \frac{\sqrt{n}\epsilon_{\mathrm{abs}}}{\left\|A^T\mathbf{y}^{(k+1)}\right\|_2} + \epsilon_{\mathrm{rel}} \; . \tag{60}$$

The most likely reason for this scaling issue not having previously been observed is that (i) in many cases the natural scaling of the problem leads to values of Eq. (55) and (56) that do not differ too greatly from unity, so that the standard residuals are roughly equal to the normalised version, and (ii) the default choice of $\mu = 10$ makes the method insensitive to smaller variations in the residual ratio. In Sec. VII this is demonstrated to be the case for the example sparse coding problem considered.

V. Adaptive Multiplier Policy

The fixed multiplier τ is a weakness of the penalty update policy Eq. (37). If τ is small, then a large number of iterations

[1]Normalisation is not mentioned at all in the original work proposing the adaptive ρ policy [6], [15]. Boyd et al. do point out [2, Sec 3.4.1] the possibility of taking $\epsilon_{\mathrm{pri}}^{(k+1)}$ and $\epsilon_{\mathrm{dua}}^{(k+1)}$ into account when attempting to balance primal and dual residuals, but this is posed as an optional enhancement rather than being critical to correct functioning of the method in the general case, as the results presented here indicate it is.

may be required[2] to reach an appropriate ρ value if $\rho^{(0)}$ is poorly chosen so that $\left\|\mathbf{r}^{(k)}\right\|_2 \gg \left\|\mathbf{s}^{(k)}\right\|_2$ or vice versa. On the other hand, if τ is large, the corrections to ρ may be too large when ρ is close to the optimal value.

A straightforward solution is to adapt τ at each iteration

$$\tau^{(k)} = \begin{cases} \sqrt{\xi^{-1}\left\|\mathbf{r}^{(k)}\right\|_2 / \left\|\mathbf{s}^{(k)}\right\|_2} & \text{if } 1 \leq \sqrt{\xi^{-1}\left\|\mathbf{r}^{(k)}\right\|_2 / \left\|\mathbf{s}^{(k)}\right\|_2} < \tau_{\max} \\ \sqrt{\xi\left\|\mathbf{s}^{(k)}\right\|_2 / \left\|\mathbf{r}^{(k)}\right\|_2} & \text{if } \tau_{\max}^{-1} < \sqrt{\xi^{-1}\left\|\mathbf{r}^{(k)}\right\|_2 / \left\|\mathbf{s}^{(k)}\right\|_2} < 1 \\ \tau_{\max} & \text{otherwise} \end{cases} , \tag{61}$$

where $\tau_{\max}$ provides a bound on τ, and ξ, allows targeting a residual ratio that differs from unity when necessary, in which case the ρ update policy must also be modified as

$$\rho^{(k+1)} = \begin{cases} \tau\rho^{(k)} & \text{if } \left\|\mathbf{r}^{(k)}\right\|_2 > \xi\mu\left\|\mathbf{s}^{(k)}\right\|_2 \\ \tau^{-1}\rho^{(k)} & \text{if } \left\|\mathbf{s}^{(k)}\right\|_2 > (\mu/\xi)\left\|\mathbf{r}^{(k)}\right\|_2 \\ \rho^{(k)} & \text{otherwise} \end{cases} . \tag{62}$$

The motivation for variable ξ will become apparent shortly, but in most cases $\xi = 1$. Since τ is bounded, the convergence results [6] still hold for this extension.

VI. BPDN

To illustrate these issues, we will focus on Basis Pursuit DeNoising (BPDN) [16],

$$\arg\min_{\mathbf{x}} \frac{1}{2} \|D\mathbf{x} - \mathbf{s}\|_2^2 + \lambda \|\mathbf{x}\|_1 , \tag{63}$$

a standard problem in computing sparse representations corresponding to Eq. (6) with

$$f(\mathbf{x}) = \frac{1}{2} \|D\mathbf{x} - \mathbf{s}\|_2^2 \quad g(\mathbf{x}) = \lambda \|\mathbf{x}\|_1$$
$$A = 1 \quad B = -1 \quad \mathbf{c} = 0 . \tag{64}$$

Solving via ADMM we have problem P

$$\arg\min_{\mathbf{x}} \frac{1}{2} \|D\mathbf{x} - \mathbf{s}\|_2^2 + \lambda \|\mathbf{z}\|_1 \text{ s.t. } \mathbf{x} = \mathbf{z} \tag{65}$$

with Lagrangian

$$L(\mathbf{x}, \mathbf{z}, \mathbf{y}) = \frac{1}{2} \|D\mathbf{x} - \mathbf{s}\|_2^2 + \lambda \|\mathbf{z}\|_1 + \mathbf{y}^T(\mathbf{x} - \mathbf{z}) . \tag{66}$$

We also consider Convolutional BPDN (CBPDN), a variant of BPDN constructed by replacing the linear combination of a set of dictionary vectors by the sum of a set of convolutions with dictionary filters [17]

$$\arg\min_{\{\mathbf{x}_m\}} \frac{1}{2} \left\| \sum_m \mathbf{d}_m * \mathbf{x}_m - \mathbf{s} \right\|_2^2 + \lambda \sum_m \|\mathbf{x}_m\|_1 , \tag{67}$$

where $\{\mathbf{d}_m\}$ is a set of M dictionary *filters*, $*$ denotes convolution, and $\{\mathbf{x}_m\}$ is a set of coefficient maps. Algebraicly, this variant is a special case of standard BPDN, so that the same scaling properties apply, but since the dictionaries in this form are very highly overcomplete (the overcompleteness factor is equal to the number of filters M), one may expect that this variant might exhibit at least somewhat different behaviour in practice. A further difference is that the $\{\mathbf{x}_m\}$ can be efficiently computed without any factorisation of system matrices [12], so in this case the penalty update policy is applied at every iteration instead of at every 10 iterations.

A. Scaling Properties

The scaling properties of the BPDN problem with respect to the scalar multiplication of the input signal $\mathbf{s}$ depend on whether the dictionary is considered to have fixed scaling or scale with the signal. The former is the more common situation since the dictionary is usually normalised, but the latter situation does occur in an *endogenous* sparse representation [18], in which the signal is also used as the dictionary (with constraints on the sparse representation to avoid the trivial solution), usually without normalisation of the dictionary.

B. Fixed Dictionary

First, define problem $\tilde{P}$ with signal $\mathbf{s}$ scaled by δ

$$\arg\min_{\mathbf{x}} \frac{1}{2} \|D\mathbf{x} - \delta\mathbf{s}\|_2^2 + \delta\lambda \|\mathbf{z}\|_1 \text{ s.t. } \mathbf{x} = \mathbf{z} , \tag{68}$$

representing the most common case in which the columns of D are normalised and D does not scale with $\mathbf{s}$. The corresponding Lagrangian is

$$\begin{aligned} \tilde{L}(\mathbf{x}, \mathbf{z}, \mathbf{y}) &= \frac{1}{2} \|D\mathbf{x} - \delta\mathbf{s}\|_2^2 + \delta\lambda \|\mathbf{z}\|_1 + \mathbf{y}^T(\mathbf{x} - \mathbf{z}) \\ &= \frac{1}{2} \left\|D\delta\delta^{-1}\mathbf{x} - \delta\mathbf{s}\right\|_2^2 + \delta\lambda \left\|\delta\delta^{-1}\mathbf{z}\right\|_1 \\ &\qquad + \mathbf{y}^T(\delta\delta^{-1}\mathbf{x} - \delta\delta^{-1}\mathbf{z}) \\ &= \delta^2 L(\delta^{-1}\mathbf{x}, \delta^{-1}\mathbf{z}, \delta^{-1}\mathbf{y}) . \end{aligned} \tag{69}$$

Comparing with Eq. (47) it is clear that we need to set

$$\alpha = \delta^2 \quad \gamma = \delta^{-1} \quad \beta = \delta \tag{70}$$

to use the ADMM scaling results of Sec. IV. In this case the scaling behaviour is such that changing δ does *not* alter the ratio of primal and dual residuals. Note that this merely implies that the adaptive penalty parameter policy with standard residuals is not *guaranteed* to fail when the signal is scaled; it does not follow that the problem scaling is such that normalised residuals are not necessary.

C. Dictionary Scales with Signal

In the second form of scaling, D is not normalised, and scales linearly with $\mathbf{s}$. In this case problem $\tilde{P}$ with signal $\mathbf{s}$ and dictionary D scaled by δ is

$$\arg\min_{\mathbf{x}} \frac{1}{2} \|\delta D\mathbf{x} - \delta\mathbf{s}\|_2^2 + \delta^2\lambda \|\mathbf{z}\|_1 \text{ s.t. } \mathbf{x} = \mathbf{z} . \tag{71}$$

[2] In many problems to which ADMM is applied, solving the $\mathbf{x}$ update Eq. (14) involves solving a large linear system, which can be efficiently achieved by pre-computing an LU or Cholesky factorization of the system matrix for use in each iteration. Since the system matrix depends on ρ, it is necessary to re-compute the factorization when ρ is updated. (This can be avoided by use of an alternative factorisation [8, Sec. 4.2], but since this method is substantially more computationally expensive in some cases, and since a thorough comparison with this alternative is beyond the scope of the present paper, it will not be considered further here.) Given the computational cost of the factorization, it is reasonable to only apply the ρ update at every 10 (for example) iterations so that the cost of the factorization can be amortized over multiple iterations. This compromise further reduces the adaption rate of the adaptive penalty policy.

The corresponding Lagrangian is

$$\begin{aligned}\tilde{L}(\mathbf{x},\mathbf{z},\mathbf{y}) &= \frac{1}{2}\left\|\delta D\mathbf{x} - \delta\mathbf{s}\right\|_2^2 + \delta^2\lambda\left\|\mathbf{z}\right\|_1 + \mathbf{y}^T(\mathbf{x}-\mathbf{z}) \\ &= \frac{\delta^2}{2}\left\|D\mathbf{x} - \mathbf{s}\right\|_2^2 + \delta^2\lambda\left\|\mathbf{z}\right\|_1 \\ &\qquad + \delta^2\delta^{-2}\mathbf{y}^T(\mathbf{x}-\mathbf{z}) \\ &= \delta^2 L(\mathbf{x},\mathbf{z},\delta^{-2}\mathbf{y})\ . \end{aligned} \tag{72}$$

Comparing with Eq. (47) it is clear that we need to set

$$\alpha = \delta^2 \quad \gamma = 1 \quad \beta = 1 \tag{73}$$

to use the ADMM scaling results of Sec. IV. In this case the scaling behaviour is such that changing δ *does* alter the ratio of primal and dual residuals, and the adaptive penalty parameter policy with standard residuals is guaranteed to perform poorly for all but a restricted range of signal scaling values δ.

VII. Results

A. Random Dictionary

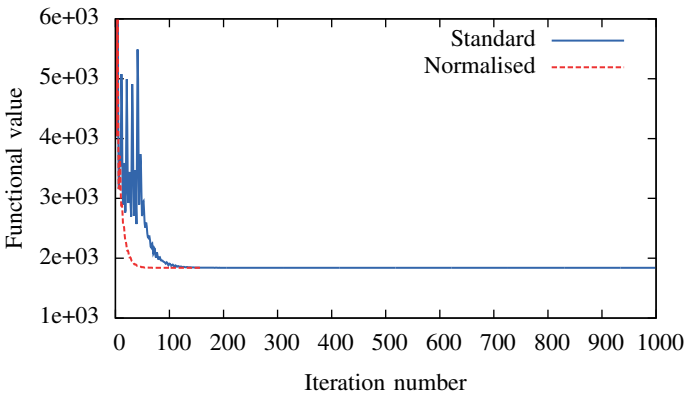


Fig. 1: A comparison of functional value evolution for the same problem with adaptive ρ based on standard and normalised residuals.

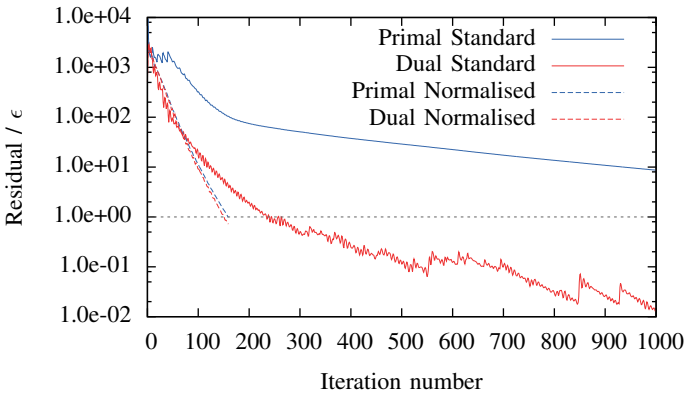


Fig. 2: A comparison of primal and dual residual evolution for the same problem with adaptive ρ based on standard and normalised residuals. For a meaningful comparison, the residuals are divided by their respective values of ϵ_{pri} or ϵ_{dua}.

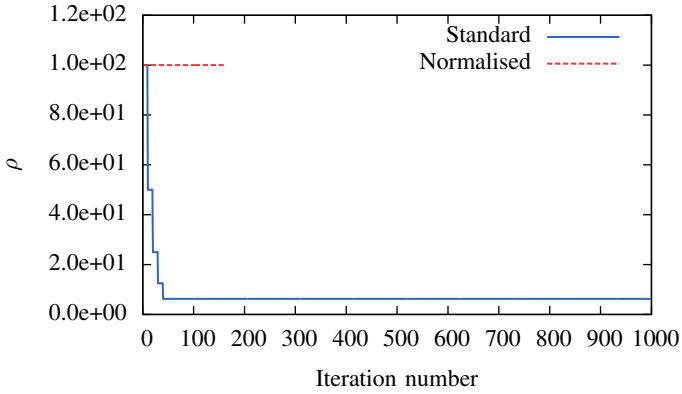


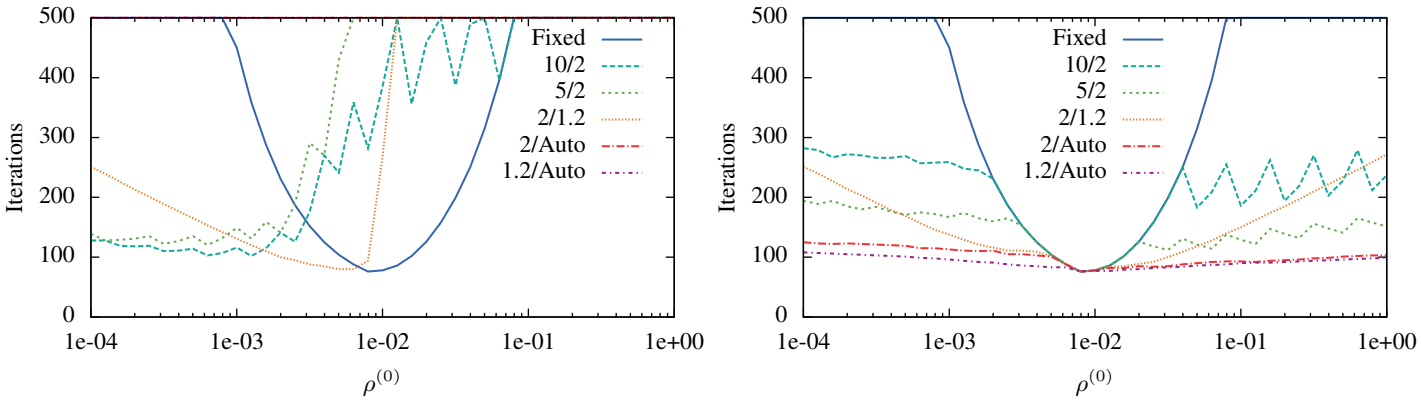
Fig. 3: A comparison of selected ρ values for the same problem with adaptive ρ based on standard and normalised residuals.

The first experiment involves sparse coefficient recovery on a random dictionary without normalisation. A dictionary $D \in \mathbb{R}^{512\times 4096}$ was generated with unit standard deviation i.i.d. entries with a Gaussian distribution, a corresponding reference coefficient vector $\mathbf{x}_0$ was constructed by assigning random values to 64 randomly selected coefficients, the remainder of which were zero, and a test signal was constructed by adding Gaussian white noise of standard deviation 0.5 to the product of D and $\mathbf{x}_0$. The experiment involves using BPDN with $\lambda = 40$ (selected for good support identification), $\epsilon_{\text{abs}} = 0$, and $\epsilon_{\text{rel}} = 10^{-4}$ to attempt to recover $\mathbf{x}_0$ from the signal, comparing performance with both standard and normalised residuals. It is clear from Figs. 1–3 that the adaptive ρ policy gives very substantially better performance with normalised residuals than with the standard definition. The desired stopping tolerance is reached within 160 iterations when using normalised residuals, but has still not been attained when the maximum iteration limit of 1000 is reached in the case of standard residuals. The performance difference is even greater if random dictionary D is generated with standard deviation greater than unity.

B. Learned Dictionary

The second set of experiments compares the performance of a fixed ρ and various adaptive ρ parameter choices, using standard and normalised residuals, for a Multiple Measurement Vector (MMV) BPDN problem. Dictionaries $D \in \mathbb{R}^{64\times 64}$, $D \in \mathbb{R}^{64\times 96}$, and $D \in \mathbb{R}^{64\times 128}$ were learned on a large training set of 8×8 image patches, and the test data consisted of 32558 zero-mean 8×8 image patches represented as a matrix $S \in \mathbb{R}^{64\times 32258}$. The number of iterations required to attain a relative stopping tolerance of $\epsilon_{\text{rel}} = 10^{-3}$ for $D \in \mathbb{R}^{64\times 128}$ and $\lambda = 10^{-2}$ is compared in Fig. 4. The following observations can be made with respect to the ability of the different methods to reduce the dependence of the number of iterations on the initial choice $\rho^{(0)}$:

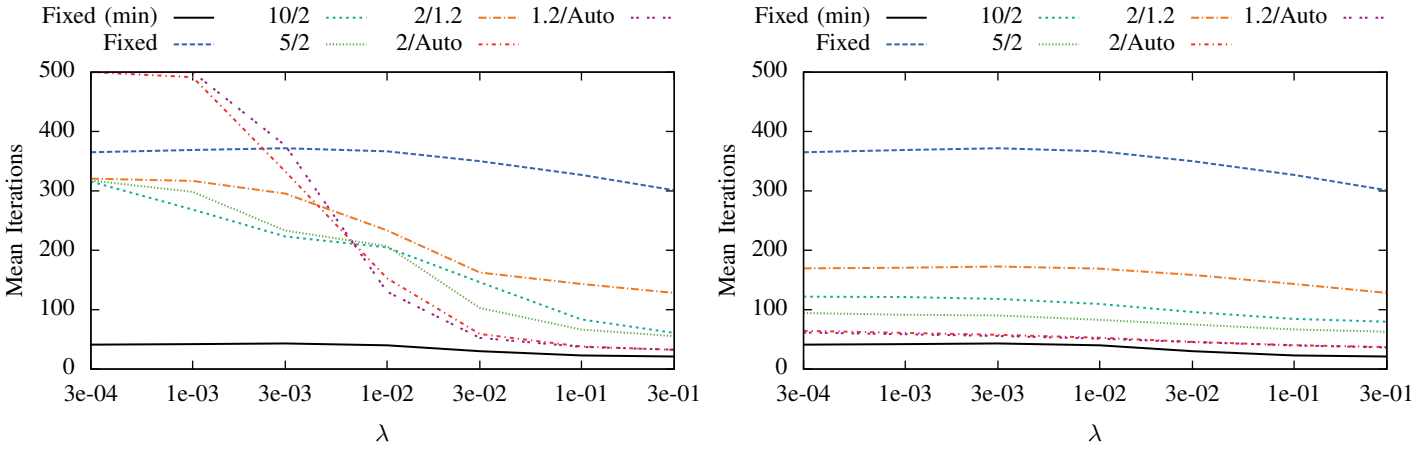
- The best choice of fixed ρ gives similar performance to the best adaptive strategy, but performance falloff is quite rapid as ρ is changed away from the optimum. Given the absence of techniques for identifying the optimum ρ



(a) Standard residuals

(b) Normalised residuals

Fig. 4: Variation with $\rho^{(0)}$ of number of iterations required to reach a relative stopping tolerance of $\epsilon_{\text{rel}} = 10^{-3}$ for different variants of the adaptive ρ policy, and for standard and normalised residuals, in a BPDN problem with $D \in \mathbb{R}^{64\times 128}$ and $\lambda = 10^{-2}$. The variant labels are "Fixed", indicating that ρ is fixed at $\rho^{(0)}$ and is not adapted, of the form μ/τ, or of the form μ/Auto, which indicates that τ is adapted as in Eq. (61), with $\tau_{\max} = 100$ and $\xi = 1$.



(a) Standard residuals

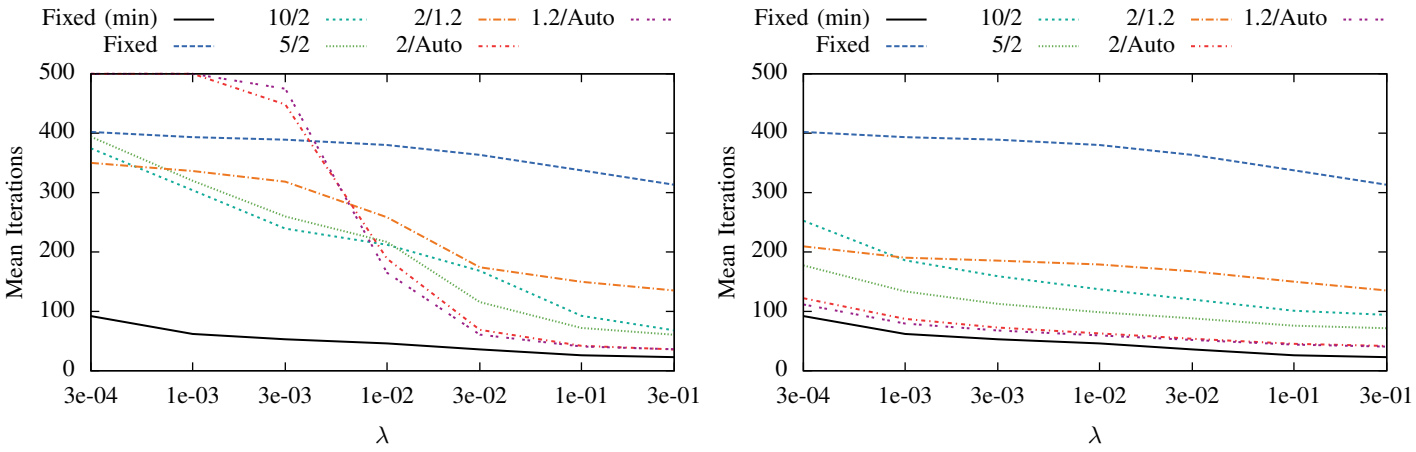
(b) Normalised residuals

Fig. 5: Mean number of iterations (averaged over all values of $\rho^{(0)}$) required to reach a relative stopping tolerance of $\epsilon_{\text{rel}} = 10^{-3}$ for different variants of the adaptive ρ policy, and for standard and normalised residuals, in a BPDN problem with $D \in \mathbb{R}^{64\times 64}$ and varying λ. The variant labels are "Fixed", indicating that ρ is fixed at $\rho^{(0)}$ and is not adapted, of the form μ/τ, or of the form μ/Auto, which indicates that τ is adapted as in Eq. (61), with $\tau_{\max} = 100$ and $\xi = 1$. "Fixed (min)" denotes the minimum number of iterations (i.e. not the mean) obtained via the best fixed choice of $\rho^{(0)}$ at each value of λ.

a priori for most problems, it is clear that the adaptive strategy can play a valuable role in reducing computation time.

- When using normalised residuals, there is an overall improvement with smaller μ. In particular, it appears that, at least for the BPDN problem, the standard choice of $\mu = 10$ is too coarse, and benefit can be obtained from finer control of the residual ratio,
- When using standard residuals, the converse is true, performance decreasing with smaller μ. This should not be surprising given the previously identified theoretical problems regarding the use of standard residuals in Eq. (37): the errors in the residual ratio that are masked by setting $\mu = 10$ become increasingly apparent as μ is reduced in an attempt at exerting finer control over the residual ratio. In this case the performance of the adaptive τ methods based on Eq. (61) is particularly poor because the adaptive τ allows ρ to be more rapidly adjusted to the incorrect value based on the incorrect residual ratios.
- The best overall performance is provided by the two automatic τ methods based on Eq. (61) with normalised residuals.

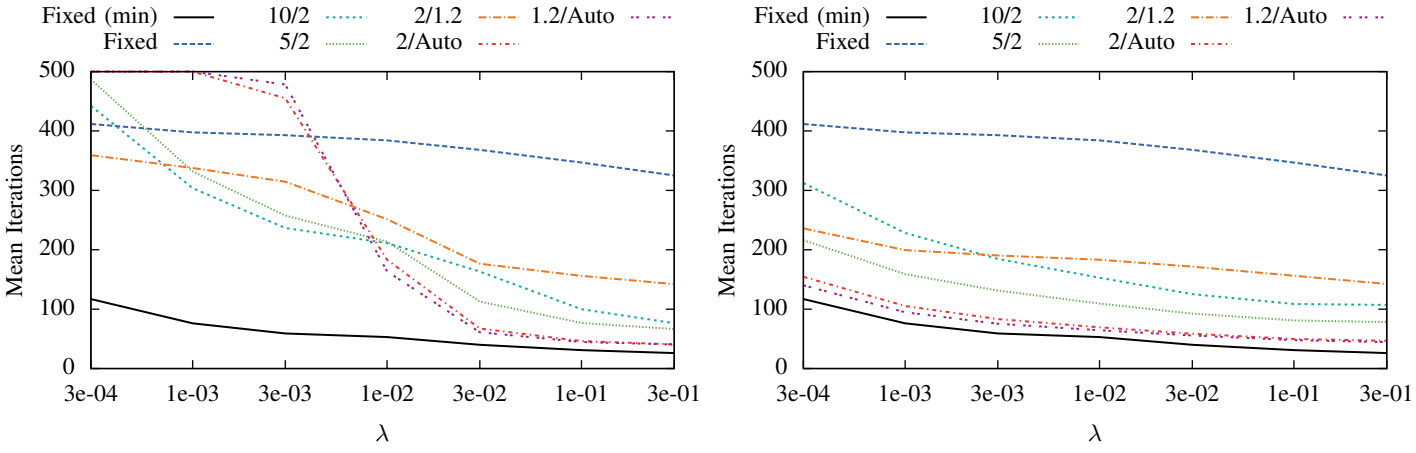
Comparisons of the different strategies over a wide range of λ values and three different dictionary sizes are presented



(a) Standard residuals

(b) Normalised residuals

Fig. 6: Mean number of iterations (averaged over all values of $\rho^{(0)}$) required to reach a relative stopping tolerance of $\epsilon_{\text{rel}} = 10^{-3}$ for different variants of the adaptive ρ policy, and for standard and normalised residuals, in a BPDN problem with $D \in \mathbb{R}^{64\times 96}$ and varying λ. The variant labels are "Fixed", indicating that ρ is fixed at $\rho^{(0)}$ and is not adapted, of the form μ/τ, or of the form μ/Auto, which indicates that τ is adapted as in Eq. (61), with $\tau_{\max} = 100$ and $\xi = 1$. "Fixed (min)" denotes the minimum number of iterations (i.e. not the mean) obtained via the best fixed choice of $\rho^{(0)}$ at each value of λ.



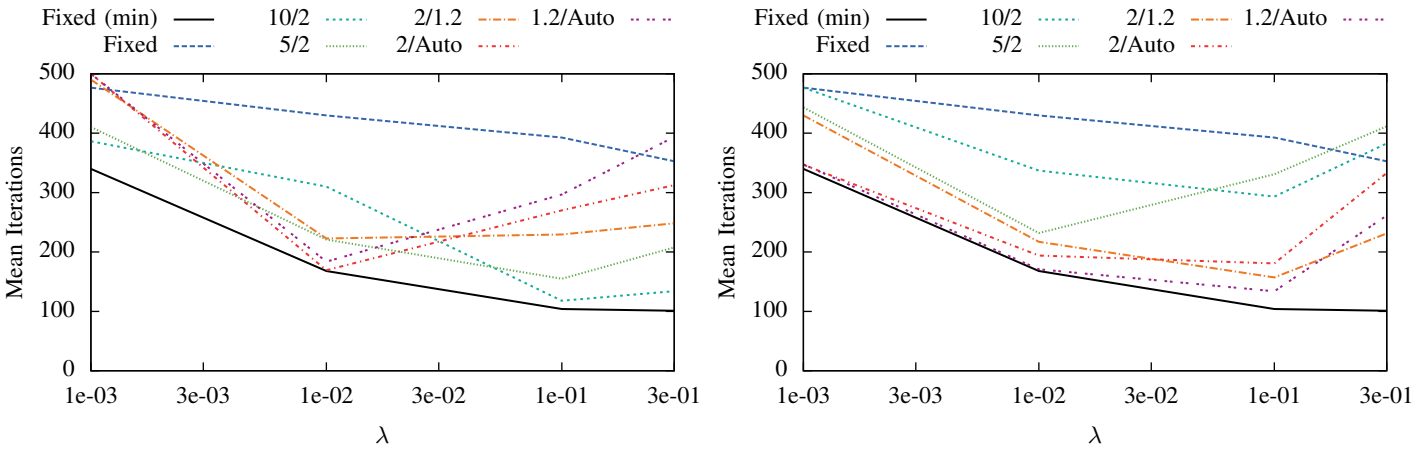
(a) Standard residuals

(b) Normalised residuals

Fig. 7: Mean number of iterations (averaged over all values of $\rho^{(0)}$) required to reach a relative stopping tolerance of $\epsilon_{\text{rel}} = 10^{-3}$ for different variants of the adaptive ρ policy, and for standard and normalised residuals, in a BPDN problem with $D \in \mathbb{R}^{64\times 128}$ and varying λ. The variant labels are "Fixed", indicating that ρ is fixed at $\rho^{(0)}$ and is not adapted, of the form μ/τ, or of the form μ/Auto, which indicates that τ is adapted as in Eq. (61), with $\tau_{\max} = 100$ and $\xi = 1$. "Fixed (min)" denotes the minimum number of iterations (i.e. not the mean) obtained via the best fixed choice of $\rho^{(0)}$ at each value of λ.

in Figs. 5–7. The mean number of iterations for all ρ values is plotted against λ, and also compared with the minimum number of iterations obtained for the best fixed choice of ρ. The most important observations to be made are:

- The standard residuals give similar performance to the normalised residuals for the larger values of λ since in this regime the normalisation quantities turn out to be close to unity.
- At smaller values of λ, the normalised residuals give much better performance.
- Considered over the entire range of λ values, the normalised residuals all give better performance than their un-normalised counterparts.
- Of the methods using normalised residuals, the adaptive τ methods based on Eq. (61) gives substantially better performance than the standard methods.



(a) Standard residuals

(b) Normalised residuals

Fig. 8: Mean number of iterations (averaged over all values of $\rho^{(0)}$) required to reach a relative stopping tolerance of $\epsilon_{\text{rel}} = 10^{-3}$ for different variants of the adaptive ρ policy, and for standard and normalised residuals, in a CBPDN problem with a $8 \times 8 \times 64$ dictionary and varying λ. The variant labels are "Fixed", indicating that ρ is fixed at $\rho^{(0)}$ and is not adapted, of the form μ/τ, or of the form μ/Auto, which indicates that τ is adapted as in Eq. (61), with $\tau_{\max} = 100$ and $\xi = 1$. "Fixed (min)" denotes the minimum number of iterations (i.e. not the mean) obtained via the best fixed choice of $\rho^{(0)}$ at each value of λ.

C. Convolutional BPDN Problem

The penalty update strategies were also compared in application to a Convolutional BPDN problem consisting of jointly computing the representations of two 256×256 pixel images[3] (the well-known "Lena" and "Barbara" images), with a dictionary consisting of 64 filters of size 8×8 samples and for a range of λ and $\rho^{(0)}$ values. It can be seen from Fig. 8 that the normalised residuals give good performance for $\lambda \leq 0.1$, but for larger values of λ neither standard nor normalised residuals provide performance close to that of the best fixed ρ.

1) Target Residual Ratio: The reason for this phenomenon is the assumption implicit in update policy Eq. (37) that the best progress is obtained when primal and dual residuals are balanced, presumably based on the notion that allowing one of the residuals to progress much faster than the other is a waste of resources if both residuals are subject to the same stopping tolerance, since the algorithm will only terminate when the more slowly reducing residual reaches this threshold. It is possible, however, that adjusting ρ so that one residual progresses faster than the other leads to an overall improvement so that the more slowly progressing residual reaches the specified threshold before either of the thresholds in the case where their progress is equalised. Such a situation, illustrated in Fig. 9, motivates the inclusion of a factor ξ in Eq. (61) and (62), allowing the update policy to target a residual ratio other than unity.

The effect of varying ξ was investigated by running a large number of computational experiments for the CBPDN problem, with a $8 \times 8 \times 64$ dictionary and for different values of λ (6 approximately logarithmically spaced values in the range 1×10^{-3} to 0.3), ρ (51 logarithmically spaced values in the range $10^{-1}\lambda$ to $10^{4}\lambda$), and ξ (21 values in the range 0.3 to 10.0). The mean and standard deviation over $\rho^{(0)}$ of the number of iterations required to reach stopping tolerance $\epsilon_{\text{abs}} = 0, \epsilon_{\text{rel}} = 10^{-3}$ are displayed in Fig. 10 and 11 respectively. It can be observed that that the value of ξ giving the minimum number of iterations varies with λ, and that considering the mean over $\rho^{(0)}$ of the number of iterations is a reasonable criterion since the variation with $\rho^{(0)}$ is small when ξ is well chosen.

Since the best ξ varies with λ, it is reasonable to ask, in the

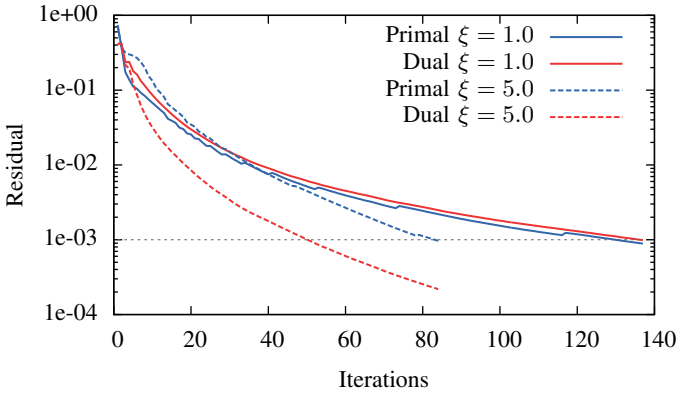


Fig. 9: Evolution of primal and dual residuals for two different choices of ξ in a CBPDN problem with an $8 \times 8 \times 32$ dictionary, $\lambda = 0.3$, and $\rho^{(0)} = 251$. The ρ update policy was as in Eq. (62), with normalised residuals, $\mu = 1.2$, and with adaptive τ as in Eq. (61), with $\tau_{\max} = 100$.

[3] As is common practice in convolutional sparse representations, the representation was computed after a highpass filtering pre-processing step, consisting in this case of application of a lowpass filter, equivalent to solving the problem $\arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{s}\|_2^2 + \lambda_L \|\nabla \mathbf{x}\|_2^2$ with $\lambda_L = 5.0$, and then subtracting the lowpass filtered images from the corresponding original images.

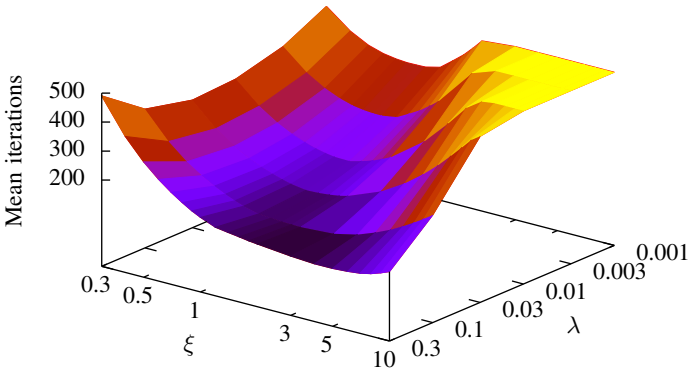


Fig. 10: Mean number of iterations, averaged over all values of $\rho^{(0)}$, against λ and ξ for a CBPDN problem with an $8 \times 8 \times 64$ dictionary. The ρ update policy was as in Eq. (62), with normalised residuals, $\mu = 1.2$, and with adaptive τ as in Eq. (61), with $\tau_{\max} = 1000$.

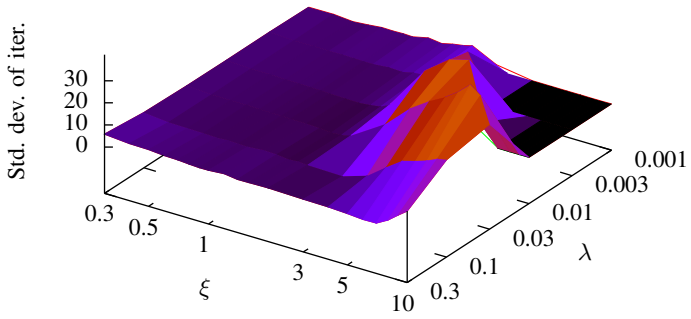


Fig. 11: Standard deviation of number of iterations with respect to $\rho^{(0)}$ in Fig. 10. Note that the variation with respect to ρ is small where the mean number of iterations is small. The standard deviation is zero for small λ and large ξ because the number of iterations is clipped to 500 by the maximum iteration limit in this region.

absence of any theory to guide the choice, whether there is a reliable way of making a good choice of ξ. By examining the data for the experiments used to generate Fig. 10 and 11, as well as for corresponding experiments with other dictionaries with 32, 96, and 128 filters of size 8×8, it was determined that the function $f(\lambda) = 1 + a^{\log_{10}(\lambda)+1}$ with $a = 18.3$ provides a reasonable fit to the best choice of ξ for each λ, over all of these dictionaries. The fit of this function to the experimental data for the dictionary of 64 filters is shown in Fig. 12a, and a corresponding performance comparison in terms of mean iterations averaged over $\rho^{(0)}$ is displayed in Fig. 12b. Note that none of the fixed choices of ξ provide good performance over the entire range of λ values, while ξ chosen according to $f(\lambda)$ gives the same performance as the best choices of ξ at each λ.

Additional experiments using different test images (the "Kiel" and "Bridge" standard images) as well as different dictionary filters sizes (12×12) indicate that $f(\lambda)$ provides a good choice of ξ over a wide range of conditions. While the choice of a giving the best fit does vary with test images, filter size, and number of filters[4] , the performance is not highly sensitive to the choice of a (note that the mean iteration surface for large λ is flat over a wide range of ξ values in Fig. 10) and the choice of $a = 18.3$ used in Fig. 12a was found to give performance at or close to the best choice of ξ in all the cases considered.

VIII. Conclusion

The scaling properties of the standard definitions of the primal and dual residuals are shown to represent a potentially serious weakness in a popular adaptive penalty strategy [6] for ADMM algorithms. The proposed solution is to normalise these residuals so that they become invariant to scalings of the ADMM problem to which the solution is also invariant. The impact of this issue is demonstrated using BPDN sparse coding as an example problem. These experiments show that the standard adaptive penalty strategy [6] performs very poorly in certain cases, while the proposed modification based on normalised residuals is more robust.

It is also shown via computational experiments involving the BPDN problem that it is possible to improve quite substantially on the standard choices of $\mu = 10$ and $\tau = 2$, and that the proposed adaptive choice of τ provides further performance improvements. For highly overcomplete dictionaries, such as those that occur in Convolutional BPDN, the use of a target residual ratio ξ other than unity is shown to improve performance for certain values of λ. While it is not guaranteed that these effects will generalise to all ADMM problems, the performance improvements obtained for the BPDN problem suggest that they deserve consideration where the adaptive penalty strategy has been applied to other types of ADMM problem.

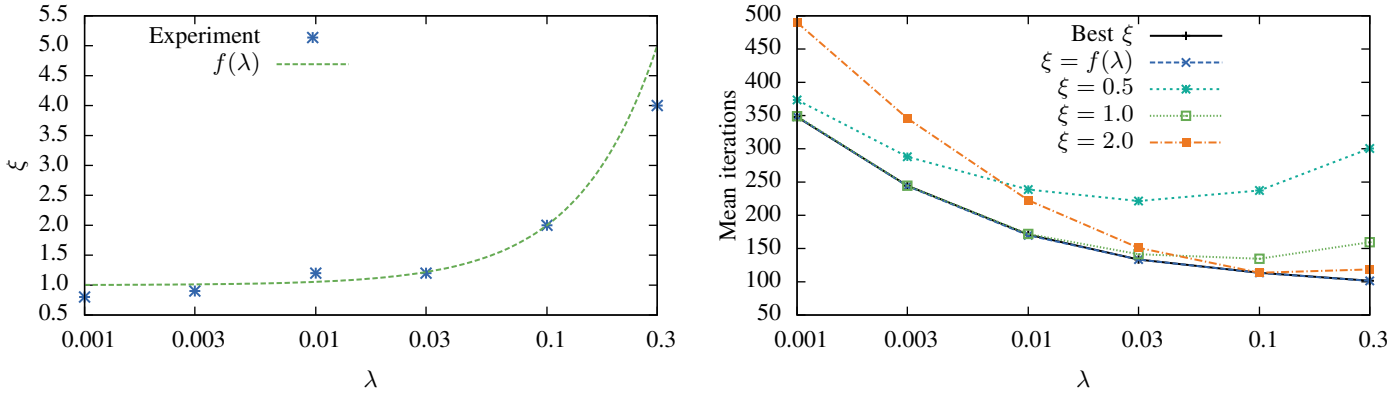
In the interests of reproducible research, software implementations of the main algorithms proposed here are made publicly available [19].

References

[1] T. Goldstein and S. J. Osher, "The split Bregman method for l1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009. doi:10.1137/080725891

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010. doi:10.1561/2200000016

[4] The importance of selecting $\xi > 1$ for larger λ values appears to be related to dictionary overcompleteness, corresponding to the number of filters for the CBPDN problem. It is also the case for the standard BPDN problem that the best choice of ξ is greater than unity for larger λ values, but for the much lower ovecompleteness ratios usually encountered in this problem variant, the performance effect is far smaller, and the loss in choosing fixed $\xi = 1.0$ is usually negligible.



(a) Function fit to best values of ξ for different λ.

(b) Mean iterations for different values of λ.

Fig. 12: (a) shows the good fit of function $f(\lambda) = 1 + 18.3^{\log_{10}(\lambda)+1}$ to the values of ξ that minimise the mean (over all values of $\rho^{(0)}$) number of required iterations for different values of λ, determined by running a large number of simulations for different values of ξ, ρ, and λ. (b) shows the variation with λ of the mean (over all values of $\rho^{(0)}$) number of iterations for the best choice of ξ as in (a), for ξ chosen according to the function $f(\lambda)$, and for three fixed choices of ξ. All simulations were for a CBPDN problem with a $8 \times 8 \times 64$ dictionary and $\epsilon_{\text{abs}} = 0, \epsilon_{\text{rel}} = 10^{-3}$. The ρ update policy was as in Eq. (62), with normalised residuals, $\mu = 1.2$, and with adaptive τ as in Eq. (61), with $\tau_{\max} = 1000$.

[3] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "An Augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 681–695, Mar. 2011. doi:10.1109/tip.2010.2076294

[4] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems," arXiv, Tech. Rep. arXiv:1306.2454, 2014, preprint of manuscript submitted to IEEE Transactions on Automatic Control.

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

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

[7] A. Hansson, Z. Liu, and L. Vandenberghe, "Subspace system identification via weighted nuclear norm optimization," *CoRR*, vol. abs/1207.0023, 2012. [Online]. Available: http://arxiv.org/abs/1207.0023

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

[9] V. Q. Vu, J. Cho, J. Lei, and K. Rohe, "Fantope projection and selection: A near-optimal convex relaxation of sparse PCA," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., 2013, pp. 2670–2678.

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

[11] D. S. Weller, A. Pnueli, O. Radzyner, G. Divon, Y. C. Eldar, and J. A. Fessler, "Phase retrieval of sparse signals using optimization transfer and ADMM," *Proc. IEEE Intl. Conf. on Image Processing*, pp. 1342–6, 2014.

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

[13] J. Eckstein, "Augmented Lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results," Rutgers Center for Operations Research, Rutgers University, Rutcor Research Report RRR 32-2012, December 2012. [Online]. Available: http://rutcor.rutgers.edu/pub/rrr/reports2012/32_2012.pdf

[14] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*. Springer, 2004.

[15] S.-L. Wang and L. Z. Liao, "Decomposition method with a variable parameter for a class of monotone variational inequality problems," *Journal of Optimization Theory and Applications*, vol. 109, pp. 415–429, 2001. doi:10.1023/a:1017522623963

[16] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998. doi:10.1137/S1064827596304010

[17] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Conf. Comp. Vis. Pat. Recog. (CVPR)*, Jun. 2010, pp. 2528–2535. doi:10.1109/cvpr.2010.5539957

[18] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk, "Greedy feature selection for subspace clustering," *Journal of Machine Learning Research*, vol. 14, pp. 2487–2517, 2013. [Online]. Available: http://jmlr.org/papers/v14/dyer13a.html

[19] B. Wohlberg, "SParse Optimization Research COde (SPORCO)," Matlab library available from http://math.lanl.gov/~brendt/Software/SPORCO/, 2014, version 0.01 **This library is being prepared for public release, and will be made available before publication of this manuscript.**

PLACE PHOTO HERE

Brendt Wohlberg received the BSc(Hons) degree in applied mathematics, and the MSc(Applied Science) and PhD degrees in electrical engineering from the University of Cape Town, South Africa, in 1990, 1993 and 1996 respectively. He is currently a staff scientist in Theoretical Division at Los Alamos National Laboratory, Los Alamos, NM. His research interests include sparse representations, exemplar-based methods for image restoration, signal and image processing inverse problems, and machine learning.