

## FIXED-POINT CONTINUATION APPLIED TO COMPRESSED SENSING: IMPLEMENTATION AND NUMERICAL EXPERIMENTS\*

Elaine T. Hale Wotao Yin and Yin Zhang  
*Department of Computational and Applied Mathematics*  
*Rice University, Houston, Texas, 77005, USA*  
*Email: Elaine.Hale@nrel.gov, wotao.yin@rice.edu, zhang@caam.rice.edu*

### Abstract

Fixed-point continuation (FPC) is an approach, based on operator-splitting and continuation, for solving minimization problems with  $\ell_1$ -regularization:

$$\min \|x\|_1 + \bar{\mu}f(x).$$

We investigate the application of this algorithm to compressed sensing signal recovery, in which  $f(x) = \frac{1}{2}\|Ax - b\|_M^2$ ,  $A \in \mathbb{R}^{m \times n}$  and  $m \leq n$ . In particular, we extend the original algorithm to obtain better practical results, derive appropriate choices for  $M$  and  $\bar{\mu}$  under a given measurement model, and present numerical results for a variety of compressed sensing problems. The numerical results show that the performance of our algorithm compares favorably with that of several recently proposed algorithms.

*Mathematics subject classification:* 65K05, 90C06, 90C25, 90C90

*Key words:*  $\ell_1$  regularization, Fixed-point algorithm, Continuation, Compressed sensing, Numerical experiments.

## 1. Introduction

The fixed-point continuation (FPC) algorithm proposed in [40] can be used to compute sparse solutions for under-determined linear systems  $Ax = b$  using the weighted least-squares formulation

$$\min_{x \in \mathbb{R}^n} \|x\|_1 + \frac{\bar{\mu}}{2} \|Ax - b\|_M^2, \quad (1.1)$$

where  $A \in \mathbb{R}^{m \times n}$ ,  $m < n$ ,  $\|p\|_M^2 = p^T M p$  and  $M$  is positive definite. This paper describes implementation details and usage guidelines for this setting, and summarizes a series of numerical experiments. The experiments simulate compressed sensing applications and provide for direct comparison of FPC with three other state-of-the-art compressed sensing recovery algorithms: GPSR [36], *l1\_ls* [42], and StOMP [27].

### 1.1. Background

In some applications, sparse solutions, that is, vectors that contain many zero elements, are preferred over dense solutions that are otherwise equally suitable. This was recognized early in geophysics, where sparse spike train signals are often of interest and data may include large sparse errors [18, 46, 60, 63]. The signal processing community seeks sparse vectors so as to

---

\* Received March 18, 2009 / Revised version received May 24, 2009 / Accepted June 6, 2009 /  
Published online December 21, 2009 /

describe a signal with just a few waveforms; similarly, statisticians often want to identify a parsimonious set of explanatory variables [17, 28, 50, 51, 64].

A direct, but computationally intractable, method for finding the sparsest solution to an under-determined linear system is to minimize the so-called “ $\ell_0$ -norm”, that is, the number of nonzeros in a vector. On the other hand, minimizing or bounding  $\|x\|_1$  has long been recognized as a practical substitute, as the two are equivalent under suitable conditions. This yields the so-called basis pursuit problem [17]

$$\min_{x \in \mathbb{R}^n} \{ \|x\|_1 \mid Ax = b \}. \quad (1.2)$$

If the “observation”  $b$  is contaminated with noise  $\epsilon$ , i.e.,

$$b = Ax + \epsilon,$$

then an appropriate norm of the residual  $Ax - b$  should be minimized or constrained. Such considerations yield a family of related optimization problems. For instance, if the noise is Gaussian then the  $\ell_1$ -regularized least squares problem

$$\min_{x \in \mathbb{R}^n} \|x\|_1 + \frac{\bar{\mu}}{2} \|Ax - b\|_2^2, \quad (1.3)$$

would be appropriate, as would the Lasso problem [64]

$$\min_{x \in \mathbb{R}^n} \{ \|Ax - b\|_2^2 \mid \|x\|_1 \leq t \}, \quad (1.4)$$

which is equivalent to (1.3) given appropriate constants  $\bar{\mu}$  and  $t$ . Note that formulations (1.1) and (1.3) are equivalent since a weighting matrix  $M$  can be incorporated in (1.3) by multiplying  $A$  and  $b$  on the left by  $M^{1/2}$ . We use the explicitly weighted formulation (1.1) because it arises naturally from the stochastic measurement model introduced in Section 3.1.

## 1.2. $\ell_1$ -Regularization and Compressed Sensing

*Compressed Sensing* is the name assigned to the idea of encoding a large sparse signal using a relatively small number of linear measurements, and decoding the signal either through minimizing the  $\ell_1$ -norm (or its variants) or employing a greedy algorithm. The current burst of research in this area is traceable to new results reported by Candes *et al.* [12–14], Donoho *et al.* [25, 26, 68] and others [59, 65]. Applications of compressed sensing include compressive imaging [62, 73, 74], medical imaging [48], multi-sensor and distributed compressed sensing [3], analog-to-information conversion [43–45, 67], and missing data recovery [81]. Compressed sensing is attractive for these and other potential applications because one can obtain a given quantity of information with fewer measurements in exchange for some additional post-processing.

In brief, compressed sensing theory shows that a sparse signal of length  $n$  can be recovered from  $m < n$  measurements by solving an appropriate variant of (1.2), (1.3), (1.4), etc., provided that the  $m \times n$  measurement matrix  $A$  possesses certain “good” properties. To date, random matrices and matrices whose rows are taken from certain orthonormal matrices have been proven to be “good”. These matrices are invariably large and dense, which contradicts the usual assumption of optimization solvers that large-scale problems appear with sparse data. The size and density of the data involved further suggest that solution algorithms should not require large linear system solves or matrix factorizations, and should take full advantage of available fast transforms like FFT and DCT. Thus it is necessary to develop dedicated algorithms for compressed sensing signal reconstruction that have the aforementioned properties and are as fast and memory-efficient as possible.

### 1.3. Several Recent Algorithms

Several recent algorithms can efficiently solve (1.3) or variants of it with large-scale data. The authors of GPSR [36], reformulate the problem as a box-constrained quadratic program, to which they apply the gradient projection method with Barzilai-Borwein steps. The algorithm  $\ell_1\text{-}\ell_s$  [42] was developed for an  $\ell_1$  regularization problem equivalent to (1.3). The authors apply an interior-point method to a log-barrier formulation of (1.3), and accelerate the computation by an efficient pre-conditioner. SPGL1 [71] applies an iterative method for solving the LASSO problem (1.4), in which an increasing sequence of  $t$ -values is used to accelerate the computation. In [52], Nesterov proposes an accelerated multistep gradient method with an error convergence rate  $\mathcal{O}(1/k^2)$ . Under some conditions, the greedy approaches including OMP [66], StOMP [27], Gradient Pursuits [9], and many of their improvements can also quickly solve (1.3). Bregman algorithms [11, 55, 78] based on [54] can quickly solve the constrained problem (1.2).

A widely used method by many researchers to solve (1.3) or the general  $\ell_1$ -minimization problems of the form:

$$\min_u \|u\|_1 + \bar{\mu}H(u) \quad (1.5)$$

for convex and differentiable functions  $H(\cdot)$  is an iterative procedure based on shrinkage (also called soft thresholding; see Eq. (2.3) in Section 2). It was independently proposed and analyzed in [34, 53] under the expectation-minimization framework for wavelet-based deconvolution, in [24] for wavelet inversion, in [2] using an auxiliary variable and the idea from the projection method [15], in [29, 30] for sparse representation and other related problems, in [22] through an optimization transfer technique, in [19] using operator-splitting, in [39] also using operator-splitting combined with a continuation technique, in [21] through an implicit PDE approach, and others. More recent algorithmic developments in this line include a two-step method TwIST [7], a generalize method SpRSA [76], as well as a recent active-set method FPC\_AS [75]. In addition, related applications and algorithms can be found in [38, 49, 77] for compressed sensing based image reconstruction, [61] for  $\ell_1$ -regularized logistic regression, [1] for image sparse representation, [6] for wavelet-based image deconvolution using a Gaussian scale mixture model, [8] for solving a cardinality constrained least-squares problem, [16, 31] for image denoising, [23] for a direct and accelerated projected gradient method, [32] for sparse representation-based image deconvolution, [35] for image deconvolution based on a bound optimization, [33] for wavelet-based image denoising using majorization-minimization algorithms, and [58] for image coding. There are also iterative coordinate descent algorithms [10, 37] and block coordinate descent algorithms [69, 70, 79], which successively minimizing the objective function with respect to a single component or a block of components. These works developed or used algorithms that are either based on or related to the iterative scheme

$$u^{k+1} \leftarrow \arg \min_u \|u\|_1 + \frac{\bar{\mu}}{2\tau^k} \|u - (u^k - \tau^k \nabla H(u^k))\|^2 \quad (1.6)$$

(or with  $\|u\|_1$  replaced by other  $\ell_1$ -related terms such as total variation) for  $k = 0, 1, \dots$  starting from a certain point  $u^0$ . The parameter  $\tau^k$  is positive and serves as the step size at iteration  $k$ . Since the unknown variable  $u$  is component-wise separable in problem (1.6), each of its components  $u_i$  can be independently obtained by the shrinkage operation (2.3), which is also referred to as soft thresholding.

Among the several approaches giving (1.6), one of the easiest ones is the following: first,  $H(u)$  is approximated by its first-order Taylor expansion at  $u^k$ , which is  $H(u^k) + \langle \nabla H(u^k), u - u^k \rangle$ ; then, since this approximation is accurate only for  $u$  near  $u^k$ ,  $u$  must be made close to  $u^k$

so an  $\ell_2$ -penalty term  $\|u - u^k\|^2/(2\tau^k)$  is added to the objective; the resulting step is

$$u^{k+1} \leftarrow \arg \min_u \|u\|_1 + \bar{\mu} \left( H(u^k) + \langle \nabla H(u^k), u - u^k \rangle + \frac{1}{2\tau^k} \|u - u^k\|^2 \right), \quad (1.7)$$

which is equivalent to (1.6) because their objectives differ by only a constant. It is easy to see that the larger the  $\tau^k$ , the larger the allowable distance between  $u^{k+1}$  and  $u^k$ . It was proved in [39] that  $\{u^k\}$  given by (1.6) converges to an optimum of (1.3) at a  $q$ -linear<sup>1)</sup> rate under certain conditions on  $H$  and  $\tau^k$ . Under weaker conditions, they also established the  $r$ -linear convergence of  $\{u^k\}$  based on the previous work by [57], and [47] on gradient projection methods. Furthermore, a new result from [39] is that the support (i.e.,  $\{i : u_i^k \neq 0\}$ ) and signs of  $u^k$  converge finitely; that is, there exists a finite number  $K$  such that  $\text{sgn}(u^k) \equiv \text{sgn}(u_{\text{opt}})$ ,  $\forall k > K$ , where  $u_{\text{opt}}$  denotes the solution of (1.3); however, an estimate or bound for  $K$  is not known. This result is reviewed in more detail in Section 2 below.

#### 1.4. Notation and Organization

For simplicity, we let  $\|\cdot\| := \|\cdot\|_2$ , the Euclidean norm. The weighted Euclidean norm corresponding to positive definite matrix  $M$  is  $\|\cdot\|_M := ((\cdot)^\top M(\cdot))^{1/2}$ . The *support* of  $x \in \mathbb{R}^n$  is  $\text{supp}(x) := \{i : x_i \neq 0\}$ . We use  $g$  to denote the gradient of the least squares term,  $f$ , and  $H$  to denote its Hessian, that is,

$$\begin{aligned} f(x) &= \|Ax - b\|_M^2/2 \\ g(x) &= A^\top M(Ax - b) \\ H &= A^\top MA, \end{aligned}$$

and these quantities simplify to  $f(x) = \|Ax - b\|^2/2$ ,  $g(x) = A^\top(Ax - b)$ , and  $H = A^\top A$  when  $M = I$ .

For any index set  $I \subseteq \{1, \dots, n\}$  (later, we will use index sets  $E$  and  $L$ ),  $|I|$  is the cardinality of  $I$ , and for any matrix  $A$ ,  $A_I$  is the sub-matrix of  $A$  consisting of the columns of  $A$  whose indices are in  $I$ . The set of solutions of Problem 1.1, or as a special case, Problem 1.3, is denoted by  $X^*$ . The signum multifunction of  $t \in \mathbb{R}$  is

$$\text{SGN}(t) := \partial|t| = \begin{cases} \{+1\} & t > 0, \\ [-1, 1] & t = 0, \\ \{-1\} & t < 0, \end{cases}$$

which is also the subdifferential of  $|t|$ . The normal distribution with zero mean and variance  $\sigma^2$  is denoted by  $N(0, \sigma^2)$ .

The paper is organized as follows. In Section 2, we summarize a fixed-point iteration scheme, which is also known as soft-shrinkage iterations, for solving (1.1). Our continuation algorithm, which extends the fixed-point iterations, is described along with other implementation details in Section 3. Section 4 describes the extensive numerical experiments conducted to demonstrate our algorithm and compare it to three other state-of-the-art algorithms in the context of compressed sensing. We summarize the experimental results in Section 5, and conclude in Section 6.

<sup>1)</sup>  $q$  stands for ‘‘quotient’’;  $\{x^k\}$  converges to  $x^*$   $q$ -linearly if  $\limsup \|x^{k+1} - x^*\|/\|x^k - x^*\| < 1$ .

## 2. Fixed-Point Iterations

### 2.1. Basic Formulation

The basic update formula of the FPC algorithm (see Algorithm 3.2, Steps 10 and 11) at the  $k^{\text{th}}$  iteration is

$$x^{k+1} \leftarrow s_\nu \circ h(x^k), \quad (2.1)$$

where the operators  $h$  and  $s_\nu$  perform gradient descent on  $f(x)$  and shrinkage on  $h(x^k)$ , respectively:

$$h(\cdot) = (\cdot) - \tau g(\cdot) \quad (2.2)$$

$$s_\nu(\cdot) = \text{sgn}(\cdot) \odot \max\{|\cdot| - \nu, \mathbf{0}\}, \quad (2.3)$$

and the parameter  $\nu$  is defined as

$$\nu = \frac{\tau}{\bar{\mu}} > 0. \quad (2.4)$$

Several approaches lead to the iterations (2.1). A forward-backward operator splitting derivation, and references to other relevant work, is given in [40]. It is easy to show that  $\tau < 2/\lambda_{\max}(H)$  is required for convergence. In addition, we have the optimality condition

$$-\bar{\mu}g_i(x^*) \in \partial|x_i^*|, \quad (2.5)$$

where  $\partial|x_i^*|$  is the subdifferential of  $|\cdot|$  at  $x_i^*$ .

For the remainder of this paper, we define

$$\hat{\lambda}_{\max} = \lambda_{\max}(A^\top MA), \quad (2.6)$$

and note that the requirement  $0 < \tau < 2/\hat{\lambda}_{\max}$  can be simplified to  $0 < \tau < 2$  by dividing  $A$  and  $b$  by  $\sqrt{\hat{\lambda}_{\max}}$ .

### 2.2. Theory

As shown in [40], the iterations (2.1) have several nice convergence properties. Although the non-smooth problem (1.1) may have more than one solution,  $g(x)$  is constant at all optimal solutions, that is, there is a vector  $g^*$  such that  $g(x) = g^*$ ,  $\forall x \in X^*$ . Thus the first order optimality conditions become  $-\bar{\mu}g^* \in \text{SGN}(x)$ , which implies that  $\bar{\mu}|g^*| \leq \mathbf{1}$  and the index set  $\{1, \dots, n\}$  can be partitioned into  $L$  and  $E$  according to

$$L := \{i : \bar{\mu}|g_i^*| < 1\} \text{ and } E := \{i : \bar{\mu}|g_i^*| = 1\}. \quad (2.7)$$

Let  $x^*$  be the limit of  $\{x^k\}$ . The following hold for all but at most finitely many iterations:

$$x_i^k = x_i^* = 0, \quad \forall i \in L, \quad (2.8)$$

$$\text{sgn}(h_i(x^k)) = \text{sgn}(h_i(x^*)) = -\bar{\mu}g_i^*, \quad \forall i \in E. \quad (2.9)$$

In addition, the numbers of iterations not satisfying (2.8) and (2.9) are bounded by, respectively,

$$\|x^0 - x^*\|^2/\omega^2 \text{ and } \|x^0 - x^*\|^2/\nu^2,$$

where  $x^0$  is the starting point for the iterations (2.1),  $\nu$  is defined in (2.4), and  $\omega := \min\{\nu(1 - \bar{\mu}|g_i^*|) \mid i \in L\} > 0$ .

When we have  $\bar{\mu}|g_i^*| < 1$  for all  $i$  such that  $x_i^* = 0$  (we call such  $x^*$  *non-degenerate*), then

$$L = \{i : x_i^* = 0\} \text{ and } E = \text{supp}(x^*). \quad (2.10)$$

Combining this with convergence of the iterations (2.1), we obtain

$$x_i^* = \text{sgn}(h_i(x^*)) \max\{\text{sgn}(h_i(x^*)) - \nu, 0\} \neq 0, \quad \forall i \in E,$$

such that  $\text{sgn}(x_i^*) = \text{sgn}(h_i(x^*))$ . Thus (2.8) and (2.9) state that the signs (positive, negative, or zero) of  $x^k$  converge to those of  $x^*$  in a finite number of iterations. When  $x^*$  is degenerate (i.e.,  $x_i^* = 0$  and  $\bar{\mu}|g_i^*| = 1$  for some  $i$ ), we do not have (2.10), but instead

$$L \subset \{i : x_i^* = 0\} \text{ and } \text{supp}(x^*) \subset E. \quad (2.11)$$

Therefore, the signs of the components of  $x^k$  in  $\text{supp}(x^*)$  will still converge finitely to those of the corresponding components of  $x^*$ , but only the components in  $L$ , a subset of  $\{i : x_i^* = 0\}$ , are guaranteed to converge finitely to zero.

Once the components of  $L$  converge to zero and the signs of the components of  $E$  converge, the dimension of  $x^k$  is essentially reduced and the algorithm becomes gradient projection for a quadratic program of size  $|E|$  constrained to a quadrant of  $\mathbb{R}^{|E|}$ . This observation led us to believe that accelerating the finite convergence process by increasing  $\nu$  would accelerate the entire algorithm. Indeed, this hunch was confirmed by numerical experiments, and motivated us to develop the continuation scheme in Section 3.2.

In [40], we also studied the rate of convergence and concluded that for quadratic  $f$ ,  $x^k$  converges linearly at a rate that depends only on the partial Hessian  $H_{EE} := A_E^\top A_E$ . For the setting in this paper, namely  $f(\cdot) = \|A(\cdot) - b\|_M^2/2$ , we showed that  $\{x^k\}$  converges to  $x^*$   $r$ -linearly, and  $\{\|x^k\|_1 + \bar{\mu}f(x^k)\}$  converges to  $\|x^*\|_1 + \bar{\mu}f(x^*)$   $q$ -linearly using the results from [47, 57]. Furthermore, if  $H_{EE}$  has full rank, we can strengthen  $r$ -linear convergence to  $q$ -linear convergence. Namely, for a particular choice of  $\tau$  we find that

$$\limsup_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} \leq \frac{\kappa(H_{EE}) - 1}{\kappa(H_{EE}) + 1}, \quad (2.12)$$

where  $\kappa(H_{EE})$  is the condition number of  $H_{EE}$  that can be much smaller than that of  $H$  when  $|E|$  is small, implying that sparsity in solution can help accelerate convergence.

### 3. The FPC Algorithm

The iterations (2.1) are the core of the fixed-point continuation (FPC) algorithm proposed in [40]. Implementation details and usage guidelines are described here. Subsection 3.1 discusses the selection of  $M$  and  $\bar{\mu}$  in the presence of Gaussian noise. Subsections 3.2 and 3.3 describe two critical acceleration heuristics: continuation on  $\bar{\mu}$  and the Barzilai-Borwein step-size on  $\tau$ , respectively. Subsection 3.4 recounts an optional post-processing procedure. Finally, Subsection 3.5 summarizes computation and storage costs.

#### 3.1. Selection of $M$ and $\mu$

For the remainder of this paper, we assume a simple measurement scenario:

$$b = A(x_s + \epsilon_1) + \epsilon_2, \quad (3.1)$$

where  $\epsilon_1$  and  $\epsilon_2$  follow the distributions  $N(0, \sigma_1^2)$  and  $N(0, \sigma_2^2)$ , respectively. Equation (3.1) implies that  $Ax_s - b$  is normally distributed with zero mean and covariance  $\sigma_1^2 AA^\top + \sigma_2^2 I$ , such that

$$\text{Prob}((Ax_s - b)^\top (\sigma_1^2 AA^\top + \sigma_2^2 I)^{-1} (Ax_s - b) \leq \chi_{1-\alpha, m}^2) = 1 - \alpha, \quad (3.2)$$

where  $\chi_{1-\alpha, m}^2$  is the  $1 - \alpha$  critical value of the  $\chi^2$  distribution with  $m$  degrees of freedom. Therefore, if the solution  $x^*$  to Problem (1.1) is to approximate  $x_s$ , one should set

$$M = (\sigma_1^2 AA^\top + \sigma_2^2 I)^{-1}, \quad (3.3)$$

and find a  $\bar{\mu}$  value that implies  $\|Ax^* - b\|_M^2 \leq \chi_{1-\alpha, m}^2$ . The  $M$  estimate is well-defined (and positive definite) as long as  $\sigma_2$  is nonzero, or  $\sigma_1$  is nonzero and  $AA^\top$  is nonsingular. An appropriate estimate for  $\bar{\mu}$  is derivable from

$$\frac{n}{\bar{\mu}^2} \geq n \|A^\top M(Ax^* - b)\|_\infty^2 \geq \|A^\top M(Ax^* - b)\|_2^2 \geq \underline{\sigma}^2 \|Ax^* - b\|_M^2,$$

where the first inequality follows from the optimality condition (2.5) and

$$\underline{\sigma}^2 := \lambda_{\min}(M^{1/2}AA^\top M^{1/2}).$$

In particular,

$$\bar{\mu} = \frac{1}{\underline{\sigma}} \sqrt{\frac{n}{\chi_{1-\alpha, m}^2}} \quad (3.4)$$

is the desired quantity, which is well-defined as long as  $A$  and  $M$  are full rank.

In the context of (2.1), having a non-identity  $M$  matrix requires additional matrix-vector multiplications, or the calculation of  $M^{1/2}$  and  $M^{1/2}A$ . However, when  $\sigma_1 = 0$  or  $AA^\top = I$  it is straightforward to set  $M = I$  and subsume the noise levels  $\sigma_1$  and  $\sigma_2$  in  $\bar{\mu}$ .

### 3.2. Continuation

The convergence results presented in Section 2 suggest that large values of  $\nu = \tau/\bar{\mu}$  may improve observed convergence rates. In particular, note that the maximum number of iterations not satisfying (2.9) is inversely proportional to  $\nu^2$ , and the maximum number of iterations not satisfying (2.8) is loosely inversely proportional to this same quantity. Furthermore,  $\mathbf{0}$  is a solution to (1.1) exactly when

$$\bar{\mu} \leq \frac{1}{\|g(\mathbf{0})\|_\infty} = \frac{1}{\|A^\top Mb\|_\infty}. \quad (3.5)$$

More generally, smaller values of  $\bar{\mu}$  in (1.1) dictate sparser solutions, while large  $\bar{\mu}$ 's favor less sparse solutions. This provides a link between smaller  $\bar{\mu}$  (hence, larger  $\nu = \tau/\bar{\mu}$ ) and faster convergence for  $\{x^k\}$  since, for instance, the  $q$ -factor in (2.12) should improve with decreasing  $|E|$  via its relationship to the condition number of  $H_{EE}(x^*)$ .

For these reasons we do not apply (2.1) directly to (1.1), but instead define a series of problems (1.1), one for each  $\mu$  in a finite increasing sequence that ends with the user-supplied  $\bar{\mu}$ . The fixed-point continuation (FPC) algorithm then consists of solving these problems with the fixed-point iterations (2.1) in turn, the starting point for the next problem being the approximate solution obtained for the previous problem.

Given a current  $\mu$  value,  $\mu_k$ , and an approximate solution  $x_k^* = x^*(\mu_k)$ , consider the readily-available quantity  $g_k^* = g(x_k^*)$ . A simple algorithm for  $\mu_{k+1}$  emerges by specifying that

$$\frac{\|g_k^*\|_\infty}{\|g_{k+1}^*\|_\infty} \geq \eta \quad (3.6)$$

for some constant  $\eta > 1$  since the optimality condition (2.5) gives the simple relationship

$$\mu_{k+1} = \eta \mu_k. \quad (3.7)$$

For a related continuation algorithm based on  $\phi_k = \|Ax_k^* - b\|_M$ , see [41].

To complete the continuation algorithm, note that the first  $\mu$  value,  $\mu_1$ , can be generated by taking  $\mu_0 = 1/\|A^\top Mb\|_\infty$  and applying (3.7). The resulting basic FPC algorithm is presented in Algorithm 3.2.

**Fixed-Point Continuation (FPC)**

**Require:**  $A, M, b, \bar{\mu}$ ; constants  $\tau \in (0, 2/\hat{\lambda}_{\max})$ ;  $\eta > 1$ ;  $xtol, gtol > 0$

$$x_p = 0, \quad x = \tau A^\top M b, \quad \mu = \frac{1}{\|A^\top M b\|_\infty}$$

**if**  $\mu \geq \bar{\mu}$  **then**

**return**  $x = 0$

**end if**

$$\mu = \min\{\eta\mu, \bar{\mu}\}$$

**while**  $\mu \leq \bar{\mu}$  **do**

**while**  $\frac{\|x - x_p\|_2}{\|x_p\|_2} > xtol \sqrt{\frac{\bar{\mu}}{\mu}}$  **or**  $\mu \|g(x_p)\|_\infty - 1 > gtol$  **do**

$$x_p = x$$

$$g = A^\top M(Ax - b)$$

$$y = x - \tau g$$

$$x = \text{sgn}(y) \circ \max\left\{|y| - \frac{\tau}{\mu}, 0\right\}$$

**end while**

$$\mu = \min\{\eta\mu, \bar{\mu}\}$$

**end while**

**return**  $x$

**3.3. Enhanced FPC: BB Steps and Line Search**

The convergence results discussed in Section 2 require  $\tau < 2/\hat{\lambda}_{\max}$  to guarantee that  $h(\cdot) = (\cdot) - \tau g(\cdot)$  is non-expansive. However, in practice larger  $\tau$  values sometimes speed convergence. A set of step-lengths well known for accelerating gradient-descents are the Barzilai and Borwein (BB) steps [4], which must be accompanied by a line search to guarantee convergence unless the objective function is quadratic, strictly convex, and unconstrained [20].

BB steps significantly improve the speed of GPSR; this motivates us to propose calculating  $\tau$  following the BB formula

$$\tau = \frac{\|x^k - x^{k-1}\|^2}{(x^k - x^{k-1})^\top (g^k - g^{k-1})}. \quad (3.8)$$

With an eye towards robustness, the BB version of FPC automatically employs a non-monotone line search to choose  $\alpha$  in the update

$$x^{k+1} = x^k + \alpha (s_\nu \circ h(x^k) - x^k). \quad (3.9)$$

The line search consists of an Armijo-type step-length condition, but with a reference function value that may be larger than the current objective function value. In particular, we follow the parameterization of [80], which, depending on the value of  $\lambda$ , results in reference values  $C$  between the current objective function value ( $\lambda = 0$ ) and the average of all previous values ( $\lambda = 1$ ). The resulting algorithm is Algorithm 3.3, in which

$$f(x) = \|x\|_1 + \frac{\bar{\mu}}{2} \|Ax - b\|_M^2. \quad (3.10)$$

**FPC with BB-Steps and Non-Monotone Line Search**

**Require:**  $A, M, b, \bar{\mu}$ ; constants  $\tau_D \in (0, 2/\hat{\lambda}_{\max})$ ;  $\eta > 1$ ;  $\lambda, c, \beta \in (0, 1)$ ;  $xtol, gtol > 0$

$$x = \tau_D A^\top M b, \quad \mu = \frac{1}{\|A^\top M b\|_\infty}$$

**if**  $\mu \geq \bar{\mu}$  **then**

**return**  $x = 0$

**end if**

$$\mu = \min\{\eta\mu, \bar{\mu}\}$$

**while**  $\mu \leq \bar{\mu}$  **do**

$$g_p = 0, g = A^\top M(Ax - b)$$

$$Q = 1, C = f(x)$$

**while**  $\frac{\|x - x_p\|_2}{\|x_p\|_2} > xtol\sqrt{\frac{\bar{\mu}}{\mu}}$  **or**  $\mu\|g(x_p)\|_\infty - 1 > gtol$  **do**

**if**  $g_p \neq 0$  **then**

$$\tau = \|x - x_p\|^2 / (x - x_p)^\top (g - g_p)$$

**else**

$$\tau = \tau_D$$

**end if**

$$y = x - \tau g, x_p = x, g_p = g$$

$$x = \text{sgn}(y) \circ \max\left\{|y| - \frac{\tau}{\mu}, 0\right\}$$

$$g = A^\top M(Ax - b)$$

$$\alpha = 1$$

**while**  $f(x_p + \alpha(x - x_p)) > C + c\alpha g_p^\top (x - x_p)$  **do**

$$\alpha = \alpha\beta$$

**end while**

$$x = x_p + \alpha(x - x_p)$$

$$g = g_p + \alpha(g - g_p)$$

$$Q_p = Q, Q = \lambda Q_p + 1, C = (\lambda Q_p C + f(x))/Q$$

**end while**

$$\mu = \min\{\eta\mu, \bar{\mu}\}$$

**end while**

In practice, our algorithm can stall in the step-size while-loop. Thus, if the line search is not successful after five tries, we reset  $\tau = \tau_D$ , recompute  $y$  and  $x$ , and take  $\alpha = 1$ . This heuristic has been practically successful, and is used in the FPC-BB results reported below. The more recent algorithm available at <http://www.caam.rice.edu/~optimization/L1/fpc/> uses a more consistent formulation of the line search.

### 3.4. Post-Processing: De-biasing

If the original signal is strictly sparse, then compressed sensing signal recovery can be decomposed into two steps: (1) identifying the nonzero components of  $x$ , and (2) estimating those components. A de-biasing algorithm, for instance Algorithm 3.4 due to [36], completes Step 1 using results from an algorithm like FPC and then computes the Step 2 estimates.

**De-biasing****Require:**  $A$ ,  $b$ , and approximate solution  $x$ ; constant  $tol$ 

$$S = \{i \mid |x_i| > tol\}$$

**if**  $1 \leq |S| \leq m$  **then**

$$Z = \{1, \dots, n\} \setminus S$$

$$x_Z = 0$$

$$x_S = \arg \min_x \|A_S x - b\|_M^2$$

**end if**

If  $|S| < m$ ,  $\text{supp}(x_s) \subset S$  and  $b = Ax_s$  ( $\sigma_1 = \sigma_2 = 0$ ), then Algorithm 3.4 will exactly reconstruct  $x_s$  [27]. Under the more realistic scenario that  $b$  is only approximately equal to  $Ax_s$ , debiasing is still advantageous in many circumstances, but it is more difficult to specify an appropriate  $tol$ .

For our numerical experiments,  $tol$  is based on the minimum 2-norm solution for  $\min_x \|Ax - b\|_M^2$ ,

$$x_{LS} = A^+ b = A^\top (AA^\top)^{-1} b, \quad (3.11)$$

where  $A^+$  is the Moore-Penrose inverse of  $A$ . If we assume that  $b$  was generated according to (3.1), then the covariance

$$\text{Cov}(x_{LS}) = \sigma_1^2 A^\top (AA^\top)^{-1} A + \sigma_2^2 A^\top (AA^\top)^{-2} A. \quad (3.12)$$

To get a scalar value for  $tol$  we note that  $\|A^\top (AA^\top)^{-1} A\|_2 = 1$  and  $\|A^\top (AA^\top)^{-2} A\|_2 = \|(AA^\top)^{-1}\|_2 = 1/\underline{\sigma}^2$ . Then we use the estimate

$$tol = 3\sqrt{\sigma_1^2 + \frac{\sigma_2^2}{\underline{\sigma}^2}}. \quad (3.13)$$

**3.5. Computational Complexity**

We are now in a position to comment on the computational complexity of Algorithms 3.2, 3.3, and 3.4 in the context of compressed sensing signal reconstruction. The storage and computation requirements of FPC and FPC-BB are listed in Table 3.1 for three situations of interest. In the first two, the  $mn$  elements of  $A$  are explicitly stored and matrix-vector multiplications cost  $\mathcal{O}(mn)$  flops. The third case refers to partial fast transform  $A$  matrices identified by listing the indices of the  $m$  rows of the  $n \times n$  transform matrix (FFT, DCT, etc.) used to compute  $b$ . In this case, matrix-vector multiplications cost  $\mathcal{O}(n \log n)$ .

The dominant operation in FPC and FPC-BB is matrix-vector multiplication. When  $M$  is the identity matrix, two multiplications, one with  $A$  and the other with  $A^\top$ , are required to compute  $g$  and  $f$ ;  $M \neq I$  adds two multiplications with  $M$ . No additional matrix-vector multiplications are required by FPC-BB's line search, unless  $M \neq I$ . In this case one additional multiplication with  $M$  is required to calculate the new value of  $f$ .

Of course the total cost of Algorithms 3.2 and 3.3 depends on the number of inner iterations, a quantity that varies with  $m$ ,  $n$ ,  $k$ ,  $\bar{\mu}$ , and problem instance. For compressed sensing we found that convergence usually occurs within 1000 iterations when the original signal is sparse enough to be accurately reconstructed. Furthermore, the number of iterations is approximately independent of the signal length  $n$  for given ratios  $\delta = m/n$  and  $\rho = k/m$ .

The primary cost of debiasing is that of solving the  $m \times |S|$  least squares problem listed on line 5 of Algorithm 3.4. In the explicit  $A$  case, we directly formulate and solve the normal

Table 3.1: Storage requirements and computational complexity of Algorithms 3.2 and 3.3 for three classes of compressed sensing problems. Under the ‘‘Computation’’ column,  $\ell$  is the number of line search steps required in a given iteration of Algorithm 3.3. The number  $\ell$  is identically equal to zero for Algorithm 3.2, and most iterations of Algorithm 3.3 also have  $\ell = 0$ .

Description	Storage (array elements)	Computation (flops per iteration)
Explicit $A$ , $M = I$	$\mathcal{O}(mn)$	$\mathcal{O}(mn) + \ell\mathcal{O}(n)$
Explicit $A$ , $M \neq I$	$\mathcal{O}(mn + m^2)$	$\mathcal{O}(mn + m^2) + \ell\mathcal{O}(m^2)$
Fast transform $A$ , $M = I$	$\mathcal{O}(n)$	$\mathcal{O}(n \log n) + \ell\mathcal{O}(n)$

equations, which costs  $\mathcal{O}(m|S|)$  or  $\mathcal{O}(m^2)$  for formulation (the latter if  $M \neq I$ ) and  $\mathcal{O}(|S|^3)$  for solution (via Cholesky factorization and backsolve). We apply the iterative solver LSQR in the fast transform case, which costs  $\mathcal{O}(n \log n)$  per iteration [56].

## 4. Numerical Experiments

Several numerical experiments were performed to demonstrate FPC and FPC-BB, and to compare them to the compressed sensing reconstruction algorithms StOMP [27], GPSR [36] and *l1\_ls* [42]. This section describes the experiments, which were run in Matlab 7.3 on a Dell Optiplex GX620 with a 3.2 GHz processor and 4 GB RAM; results follow in the next section. For a more detailed exposition, see [41].

### 4.1. Problem Parameters

For all but the last set of experiments, problem data  $A$  and  $b$  satisfying (3.1) are generated by specifying the size and type of  $A$  matrix,  $k = |\text{supp}(x_s)|$ , and the standard deviations of the noise vectors,  $\sigma_1$  and  $\sigma_2$ , where  $\text{supp}(x_s)$  is generated using `randperm`; the nonzero values of  $x_s$  are set to `2*randn`. Performance statistics are often presented as a function of

$$\delta = m/n \text{ and } \rho = k/m, \quad (4.1)$$

similar to [27].

Three types of  $A$  matrices that arise in compressed sensing applications are tested. (i) *Gaussian*  $A$  refers to  $m \times n$  matrices whose elements are taken from the standard normal distribution (`A = randn(m,n)`), except when used with the StOMP algorithm. (ii) StOMP uses *Column-normalized Gaussian*  $A$  matrices instead, which have their columns scaled to unit norm. (iii) Finally, *DCT*  $A$  refers to partial discrete cosine transform matrices, which are  $m$  rows of the  $n \times n$  discrete cosine transform matrix, chosen uniformly at random. These matrices satisfy  $AA^T = I$ , and their matrix-vector products can be computed in  $\mathcal{O}(n \log n)$  time using the `dct` and `idct` functions in Matlab’s Signal Processing Toolbox.

We typically use the  $M$  and  $\bar{\mu}$  values recommended in Section 3.1. Whenever  $M \neq cI$  for some  $c$ ,  $A$  and  $b$  are replaced with  $M^{1/2}A$  and  $M^{1/2}b$ , and  $M$  is set to  $I$ , since this is computationally preferable to the explicit formulation (1.1).

### 4.2. Algorithm Parameters

Preliminary computational experiments conducted on a  $(\delta, \rho) \in [0, 1] \times [0, 1]$  grid guided the selection of default values for the FPC, StOMP, GPSR and *l1\_ls* algorithm parameters,

subject to the guideline that the non-FPC algorithms' defaults should be used if possible.

#### 4.2.1. FPC

Our experiments showed that if  $AA^\top$  is not a multiple of  $I$ , but  $M$  is, then  $\tau$  should be near 2. For all other cases,  $\tau$  should be large when  $\delta$  is moderate to small (approximately  $\delta \leq 0.4$ ) and should decay to 1 as  $\delta$  increases. In all cases,  $\tau \geq 1$  performed better than  $\tau < 1$ . Thus, for the former case we set  $\tau$  to `2-eps`, and for the latter

$$\tau = \max\{-1.665\delta + 2.665, 1.999\}. \quad (4.2)$$

The default value of  $\eta$  (the increment factor for  $\mu$ ) is set to 4. In practice, any  $\eta$ -value between 2 and 10 is acceptable. The outer iteration corresponding to  $\mu_k$  terminates when

$$\frac{\|x - x_p\|_2}{\|x_p\|_2} \leq xtol \sqrt{\frac{\bar{\mu}}{\mu_k}} \quad \text{and} \quad \mu_k \|g(x_p)\|_\infty - 1 \leq gtol.$$

The first criterion requires the last step to be small relative to  $x^k$ ; the second checks to see if complementarity holds at the current iterate. Empirically, the presence of the second condition greatly improves accuracy, but  $gtol$  should be fairly large to ensure fast convergence for most problems. We use  $xtol = 1\text{E-}4$  and  $gtol = 0.2$ .

With the BB variant (Algorithm 3.3), we set  $\lambda = 0.85$ ,  $c = 1\text{E-}3$ , and  $\beta = 0.5$  based on [80] and the Armijo line search guidelines in [5].

#### 4.2.2. StOMP, GPSR, and l1\_ls

With StOMP we use False Alarm Rate thresholding (FAR) and  $\alpha = 0.015$ . The default maximum of 10 iterations is raised to 30 to better accommodate problems with  $m > 0.6n$ .

The GPSR results reported in this paper use GPSR v3.0, which includes a basic algorithm and an algorithm with Barzilai-Borwein (BB) steps. Continuation is optional in both, and proceeds by splitting the distance between  $1/\|A^\top Mb\|_\infty$  and  $\bar{\mu}$  into five (by default) pieces of equal log-scale length. Most of the GPSR results referred to in this paper use the BB variant with continuation and line search.

GPSR's default stopping criterion exits prematurely when  $\bar{\mu}$  is large. For comparison purposes, we use GPSR's objective value stopping criterion and set the required value to that reached by FPC for the same problem. Similarly, the maximum number of iterations for both algorithms is set to 1000, and we initialize GPSR with  $x = A^\top Mb$ , the option closest to our  $x = \tau A^\top Mb$ .

The default values for the l1\_ls algorithm described in [42] work well. The l1\_ls code does not provide a mechanism to specify any  $x^0$  value other than  $\mathbf{0}$ .

### 4.3. Experiments

#### 4.3.1. Proof-of-Concept Demonstration

Figure 4.1 displays the original, noisy, recovered and de-biased signals for a particular compressed sensing problem generated as per Section 4.1 and solved with FPC. These results serve as a basic demonstration, and also highlight FPC's ability to cope with high noise levels since FPC was able to recover all of the nonzero elements significantly larger than the underlying noise. De-biasing generally improved the large nonzero estimates, but eliminated some of the smaller nonzeros recovered by FPC.

### 4.3.2. Phase Plots: Scope

Phase plots depict how reconstruction quality varies with the number of measurements,  $m$ , and sparsity level,  $k$ , all other things being equal [27]. In general, quality improves with decreasing  $k$  and increasing  $m$ ; the exact boundary between problems in which recovery is and is not achieved for a given algorithm and a given set of problems is shown using pixel intensity on a  $\rho = k/m$  versus  $\delta = m/n$  grid. Example phase plots for FPC are shown in Figure 4.2.

### 4.3.3. Regularization Parameter Dependence: Robustness

The regularization parameter  $\bar{\mu}$  directly influences reconstruction quality by dictating the balance struck between solution sparsity and fidelity to measurements. As shown in Section 3.1, the most basic heuristic concerning  $\bar{\mu}$  is that it should be chosen in inverse proportion to the noise level. But how easy is it to specify a value of  $\bar{\mu}$  that is satisfactory with regards to both reconstruction quality and algorithm performance, given that noise levels are typically unknown?

This problem is investigated by fixing the type of measurement matrix,  $n$ ,  $m$ ,  $k$ ,  $\sigma_1$  and  $\sigma_2$ , and then solving ten problems with many values of  $\bar{\mu}$ . The resulting data are analyzed by plotting average 2-norm relative error and computational time versus  $\bar{\mu}$ , along with a vertical line at the location of the  $\bar{\mu}$  value recommended by Section 3.1. This type of experiment is also an opportunity to compare FPC, GPSR and l1\_ls to each other since they all solve (1.1). Some example results are plotted in Figures 4.3 and 4.4.

### 4.3.4. Timing Studies: Speed

Given that even a greyscale  $512 \times 512$  image is a signal of length 262,144, the primary goal of every compressed sensing recovery algorithm is to quickly solve large problems. We test the speed of FPC-BB (Algorithm 3.3), GPSR-BB (with monotone line search and continuation), l1\_ls and StOMP using an experimental set-up that is identical to the regularization parameter experiments, except that now  $\bar{\mu}$  is fixed, while  $n$  is varied. Tables 4.1 and 4.2 list computational times and relative errors for a given value of  $n$ , along with an estimate of how computational time scales with  $n$ . Namely, we assume the model CPU Time =  $Cn^\alpha$  and list a least-squares estimate for  $\alpha$  calculated from CPU time results obtained for a set of  $n$  values.

Because it is expensive to compute  $M^{1/2}$  and the largest and smallest eigenvalues of  $M^{1/2}A A^T M^{1/2}$  when  $M \neq cI$  for some  $c$ , we sometimes list timing statistics for data generation alongside those for signal recovery. Table 4.3 shows such statistics for some problems solved with the recommended  $M$  and with a simpler substitute:

$$M = (\sigma_1^2 \bar{\sigma}^2 I + \sigma_2^2 I)^{-1}, \quad (4.3)$$

where  $\bar{\sigma}^2 = (1 + \sqrt{\frac{m}{n}})^2 n$  is an upper bound approximation to  $\lambda_{\max}(AA^\top)$ .

### 4.3.5. Medical Images: Practicality

We test FPC on six images: the Shepp-Logan phantom in the Matlab Image Processing Toolbox, and five medical images in public domain. Compressed sensing problems based on these images are formulated and solved using the SPARCO toolbox [72] which, among other things,

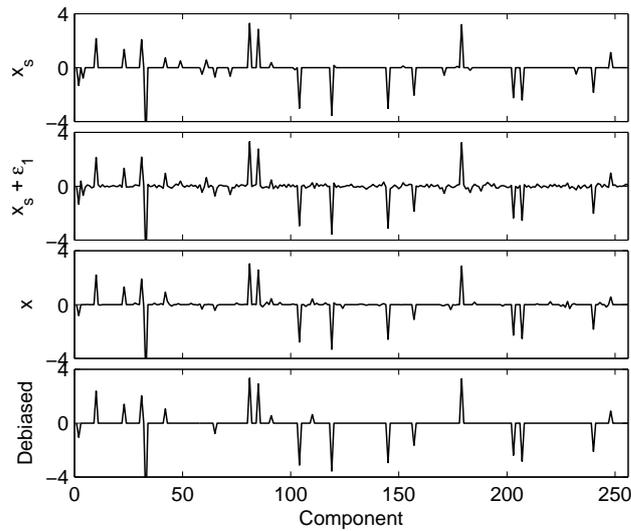


Fig. 4.1. The original, noisy, recovered, and debiased signals for a  $128 \times 256$  partial DCT  $A$  compressed sensing problem solved with FPC. The original signal has  $k = 32$  nonzeros; the noise levels were set to  $\sigma_1 = \sigma_2 = 0.1$ .

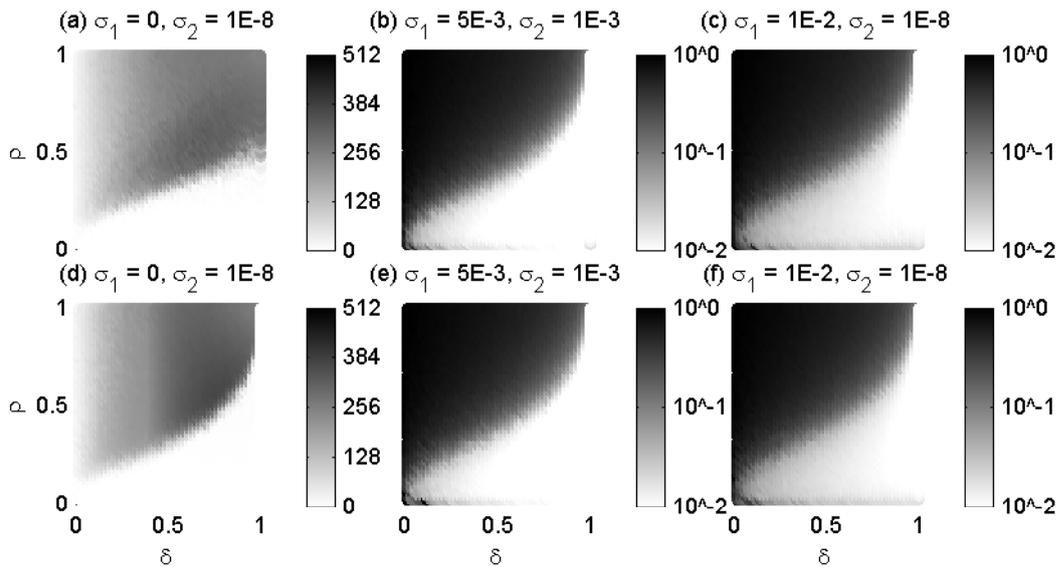


Fig. 4.2. Phase plots for FPC (Algorithm 3.2). Intensities depict average reconstruction quality (over thirty runs) as a function of  $\delta = m/n$  and  $\rho = k/m$ . In all cases,  $n = 512$  and darker shades represent poorer reconstruction. Measurement matrix type varies by row: Gaussian  $A$  on top, DCT  $A$  on the bottom. Noise scenario varies by column. In the low noise case (left column), reconstruction accuracy is measured by the number of elements not recovered to a  $1E-4$  relative error. The middle and right column plots depict relative 2-norm error,  $\|x - x_s\|/\|x_s\|$ , with errors less than  $1E-2$  shaded white.

facilitates the application of linear operators to two-dimensional signals (images) in addition to conventional one-dimensional signals (vectors).

In SPARCO, and in the discussion that follows, two-dimensional signals are vectorized before the application of any linear operators. In particular, the vectorized form of a matrix signal

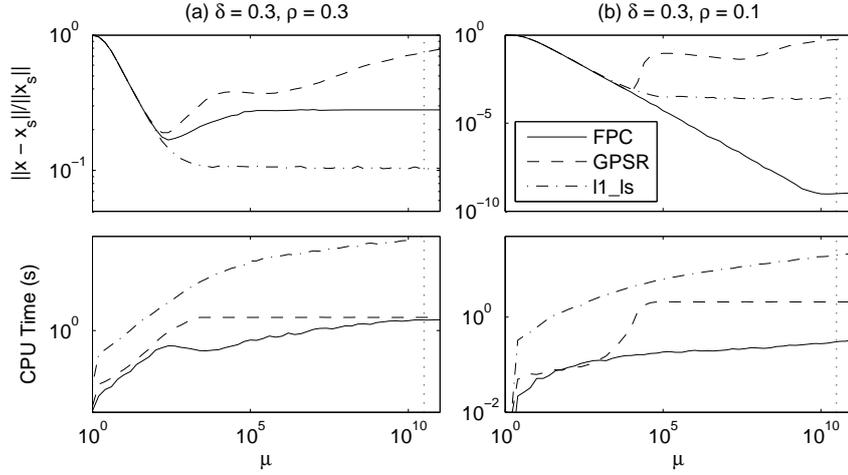


Fig. 4.3. The effects of  $\bar{\mu}$  on the accuracy and speed of FPC-BB, monotone GPSR-BB with continuation, and  $l_1$ ls for two sets of compressed sensing problems with Gaussian  $A$  matrices,  $n = 1024$ ,  $\sigma_1 = 0$  and  $\sigma_2 = 1E-8$ . As usual,  $\delta = m/n$  and  $\rho = k/m$ . The vertical dotted lines mark the recommended  $\bar{\mu}$  values (Section 3.1).

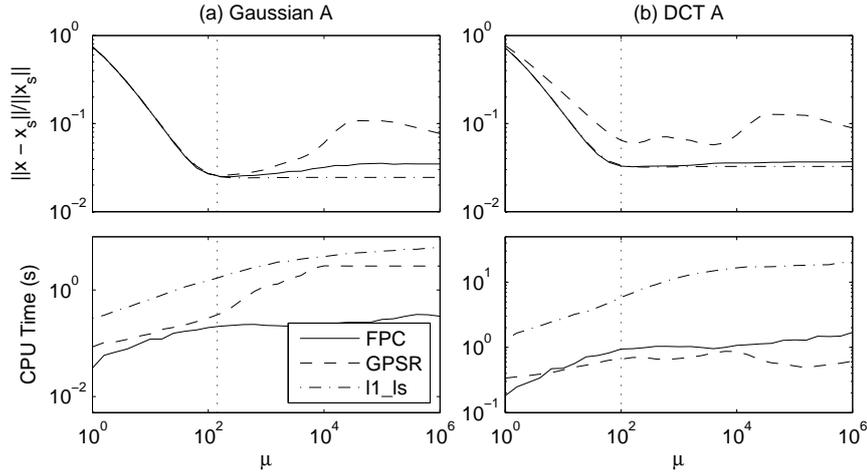


Fig. 4.4. The effects of  $\bar{\mu}$  on the accuracy and speed of FPC-BB, monotone GPSR-BB with continuation and  $l_1$ ls for two sets of compressed sensing problems with  $\sigma_1 = 1E-2$ ,  $\sigma_2 = 1E-2$ ,  $\delta = m/n = 0.5$  and  $\rho = k/m = 0.3$ : (a) Gaussian  $A$  and (b) DCT  $A$ . The vertical dotted lines mark the recommended  $\bar{\mu}$  values (Section 3.1).

$Z \in \mathbb{R}^{n \times m}$  is  $z \in \mathbb{R}^{nm}$  defined by

$$z = \text{vec}(Z) = \left( Z_1^\top \quad Z_2^\top \quad \cdots \quad Z_m^\top \right)^\top. \quad (4.4)$$

The experiments assume that the images have relatively sparse representations in Haar wavelets; that is, for the true vectorized signal  $z \in \mathbb{R}^n$ , there exists an  $x \in \mathbb{R}^n$  such that

$$z = Wx, \quad (4.5)$$

where  $W \in \mathbb{R}^{n \times n}$  is the Haar wavelet basis, and  $x$  is approximately sparse ( $x = x_s + \epsilon$ ,  $\text{supp}(x_s) \ll n$ ). We use a partial DCT measurement matrix  $A$  to compute  $b = Az$  and then

Table 4.1: Timing study results for Gaussian  $A$  matrices,  $\sigma_1 = 0$  and  $\sigma_2 = 1\text{E-}2$ . The growth of CPU time with  $n$  is reflected in the 90% confidence intervals listed for  $\alpha$ , which were calculated by applying linear least squares to estimate  $\log C$  and  $\alpha$  in the model  $\log \text{CPU time} = \log C + \alpha \log n$  (CPU time =  $Cn^\alpha$ ). The CPU times and relative errors correspond to  $n = 8192$ .

		$\delta = 0.1$			$\delta = 0.3$		
		$\alpha$	CPU (s)	Rel. Err.	$\alpha$	CPU (s)	Rel. Err.
$\rho = 0.3$	FPC	N/A			$2.1 \pm 0.1$	32.8	2.2E-1
	GPSR	N/A			$1.7 \pm 0.1$	84.4	3.6E-1
	$l1\_ls$	N/A			$1.9 \pm 0.3$	> 600	8.2E-2
	StOMP	N/A			$2.8 \pm 0.1$	174	2.8E-1
$\rho = 0.2$	FPC	$1.7 \pm 0.1$	4.41	2.8E-1	$1.8 \pm 0.3$	16.5	6.0E-4
	GPSR	$1.6 \pm 0.1$	29.5	4.1E-1	$1.7 \pm 0.1$	86.2	2.2E-1
	$l1\_ls$	$2.1 \pm 0.2$	1190	1.1E-1	$1.8 \pm 0.1$	494	1.1E-3
	StOMP	not recovered			$2.8 \pm 0.1$	205	2.2E-2
$\rho = 0.1$	FPC	$1.8 \pm 0.1$	3.62	9.4E-4	$1.8 \pm 0.2$	7.35	3.4E-4
	GPSR	$1.6 \pm 0.2$	29.3	2.0E-1	$1.8 \pm 0.2$	85.6	9.7E-2
	$l1\_ls$	$1.7 \pm 0.2$	137	9.5E-4	$1.9 \pm 0.1$	268	4.5E-4
	StOMP	not recovered			$2.7 \pm 0.1$	139	2.6E-2

try to obtain approximate wavelet coefficients  $\hat{x}$  by solving

$$\min_x \|x\|_1 + \frac{\bar{\mu}}{2} \|AWx - b\|^2 \quad (4.6)$$

with our FPC code. Finally, we complete the recovery by computing  $\hat{z} = W\hat{x}$ .

Reconstruction results are summarized in Figure 4.5, which shows the original images alongside the images recovered using  $\delta = m/n = 0.25, 0.50$ , and  $0.75$ . Quantitatively and by eye, the reconstruction quality seems good, except when  $\delta = 0.25$ . The solve times were between 1.4 to 3.5 seconds for the  $128 \times 128$  Shepp-Logan phantom image, 4.7 to 13.2 seconds for the two  $256 \times 256$  MRI images, and 20.5 to 59.2 seconds for the three  $512 \times 512$  images.

## 5. Results

The experiments described in Sections 4.3.1 and 4.3.5 show that FPC works as intended for stylized and realistic problems. As for speed in the face of large problems, the implicit storage and fast matrix multiplications associated with partial DCT matrices enabled us to solve problems with  $n = 8, 388, 608$  and  $k, m$  small in just a few minutes. Furthermore, computational time scales approximately linearly with  $n$  for all algorithms in this case. In contrast, the explicit storage of the Gaussian matrices limited those problems to  $n = 8, 192$  or less, with computation times also on the order of several minutes. StOMP's computational time in this case is between  $\mathcal{O}(n^{2.4})$  and  $\mathcal{O}(n^{2.9})$ , while the other algorithms saw dependencies between  $\mathcal{O}(n^{1.6})$  and  $\mathcal{O}(n^{2.1})$ .

Continuation was demonstrated to be a necessary component of FPC in [40]. Figure 5.1 demonstrates this for GPSR, whose basic update is similar to, but not identical with, (2.1).

Table 4.2: Timing study results for DCT  $A$  matrices,  $\sigma_1 = 1\text{E-}2$  and  $\sigma_2 = 1\text{E-}8$ . For FPC, GPSR and  $l1\_ls\_rec$  indicates that  $\bar{\mu}$  was calculated using (3.4);  $\_db$  denotes results calculated with  $\bar{\mu} = 50$  plus debiasing. See Table 4.1 for a description of  $\alpha$ . The CPU times and relative errors correspond to  $n = 131072$ . FPC, GPSR and StOMP were each able to solve at least one problem with  $n = 2097152$ .

		$\delta = 0.1$			$\delta = 0.3$		
		$\alpha$	CPU (s)	Rel. Err.	$\alpha$	CPU (s)	Rel. Err.
$\rho = 0.3$	FPC <sub>rec</sub>	N/A			$1.04 \pm 0.01$	33.9	$1.2\text{E-}1$
	FPC <sub>db</sub>	N/A			$1.02 \pm 0.01$	36.0	$7.8\text{E-}2$
	GPSR <sub>rec</sub>	N/A			$1.09 \pm 0.03$	175	$1.2\text{E-}1$
	GPSR <sub>db</sub>	N/A			$1.08 \pm 0.02$	66.7	$9.4\text{E-}2$
	$l1\_ls\_rec$	N/A			$1.03 \pm 0.04$	307	$9.7\text{E-}2$
	$l1\_ls\_db$	N/A			$1.02 \pm 0.05$	158	$1.5\text{E-}1$
	StOMP	N/A			not recovered		
$\rho = 0.2$	FPC <sub>rec</sub>	$1.01 \pm 0.02$	36.3	$1.8\text{E-}1$	$1.02 \pm 0.01$	18.3	$3.3\text{E-}2$
	FPC <sub>db</sub>	$1.01 \pm 0.01$	29.1	$1.3\text{E-}1$	$1.00 \pm 0.01$	18.7	$1.6\text{E-}2$
	GPSR <sub>rec</sub>	$1.04 \pm 0.04$	147	$1.8\text{E-}1$	$1.07 \pm 0.02$	33.4	$3.3\text{E-}2$
	GPSR <sub>db</sub>	$1.05 \pm 0.03$	41.7	$1.5\text{E-}1$	$1.03 \pm 0.02$	20.2	$2.2\text{E-}2$
	$l1\_ls\_rec$	$1.01 \pm 0.03$	483	$1.6\text{E-}1$	$1.04 \pm 0.04$	151	$3.3\text{E-}2$
	$l1\_ls\_db$	$0.96 \pm 0.04$	163	$1.5\text{E-}1$	$1.09 \pm 0.04$	103	$2.3\text{E-}2$
	StOMP	not recovered			$1.07 \pm 0.01$	61.0	$1.1\text{E-}2$
$\rho = 0.1$	FPC <sub>rec</sub>	$1.00 \pm 0.01$	16.8	$5.9\text{E-}2$	$1.01 \pm 0.01$	11.6	$2.5\text{E-}2$
	FPC <sub>db</sub>	$1.01 \pm 0.01$	12.5	$3.4\text{E-}2$	$1.00 \pm 0.01$	11.9	$1.2\text{E-}2$
	GPSR <sub>rec</sub>	$1.06 \pm 0.01$	31.3	$5.9\text{E-}2$	$1.04 \pm 0.03$	13.8	$2.5\text{E-}2$
	GPSR <sub>db</sub>	$1.03 \pm 0.03$	13.4	$3.4\text{E-}2$	$1.04 \pm 0.03$	11.5	$1.6\text{E-}2$
	$l1\_ls\_rec$	$0.98 \pm 0.02$	212	$5.8\text{E-}2$	$1.03 \pm 0.04$	99.9	$2.5\text{E-}2$
	$l1\_ls\_db$	$1.02 \pm 0.04$	114	$3.5\text{E-}2$	$1.06 \pm 0.05$	82.8	$1.6\text{E-}2$
	StOMP	not recovered			$1.09 \pm 0.01$	21.9	$1.1\text{E-}2$

Overall, some adjustments of GPSR's continuation scheme and default stopping criterion should make it competitive with FPC, but as it stands, FPC is usually more accurate and faster than GPSR. The main exception of note is problems with DCT  $A$  matrices and signal noise. As demonstrated in Figure 4.4(b), in these cases GPSR may be significantly faster than FPC, but usually does not achieve the same level of accuracy. GPSR's relative standing is typically improved by using a smaller  $\bar{\mu}$  value and de-biasing.

The regularization parameter experiments show that the recommended value of  $\bar{\mu}$  is always close to the smallest  $\bar{\mu}$  value where the relative error levels off or reaches a minimum, and that order of magnitude estimates of  $\sigma_1$  and  $\sigma_2$  are sufficient to obtain good results. In many cases, a slight overestimate of  $\bar{\mu}$  should assure good accuracy. However, overestimates are ill-advised

Table 4.3: Timing study results for Gaussian  $A$  matrices,  $\delta = 0.3$ ,  $\sigma_1 = 1\text{E-}2$  and  $\sigma_2 = 1\text{E-}2$ . See Table 4.1 for a description of  $\alpha$ . Here  $\alpha$ 's and CPU times are presented for problem generation (data) and solution (solve) using the full  $M$  (3.3) and the approximate  $M$  (4.3). Problem generation (data) includes the calculation of  $A$ ,  $b$ ,  $M$  and  $\bar{\mu}$ . The timings and relative errors correspond to  $n = 8192$ . Since StOMP does not use formulation (1.1), its results are listed arbitrarily under *Full M*.

		Full $M$					Approx. $M$				
		Data		Solve			Data		Solve		
		$\alpha$	CPU (s)	$\alpha$	CPU (s)	Rel. Err.	$\alpha$	CPU (s)	$\alpha$	CPU (s)	Rel. Err.
$\rho = 0.3$	FPC	2.9	829	1.8	11.5	1.2E-1	1.8	1.02	1.8	15.2	1.4E-1
	GPSR	"	"	1.9	50.5	1.2E-1	"	"	1.7	73.2	1.4E-1
	$l1\_ls$	"	"	1.8	128	1.0E-1	"	"	1.8	298	1.0E-1
	StOMP	1.9	1.52	not recovered			N/A				
$\rho = 0.2$	FPC	2.8	826	1.9	6.37	3.2E-2	1.7	1.02	1.9	9.07	3.6E-2
	GPSR	"	"	1.8	10.0	3.2E-2	"	"	1.8	24.6	3.6E-2
	$l1\_ls$	"	"	1.8	55.6	3.2E-2	"	"	1.7	158	3.4E-2
	StOMP	2.0	1.52	2.8	198	9.1E-1	N/A				
$\rho = 0.1$	FPC	2.9	824	1.8	4.09	2.5E-2	1.8	1.02	1.8	5.63	2.7E-2
	GPSR	"	"	1.7	4.59	2.5E-2	"	"	1.8	7.60	2.7E-2
	$l1\_ls$	"	"	1.6	39.5	2.5E-2	"	"	1.8	110	2.7E-2
	StOMP	1.8	1.52	2.9	174	4.8E-2	N/A				

in low noise situations, and should not be overdone with high noise levels, since in the former case the extra computational effort is unlikely to yield great accuracy gains, and in the latter the resulting solution may be too dense compared to  $x_s$ .

The results in Table 4.3 support the use of (4.3) as an effective, less costly substitute for the value of  $M$  recommended in Section 3.1 when that  $M$  is not a multiple of the identity matrix. While the FPC, GPSR and  $l1\_ls$  solve times are significantly longer with the approximate, rather than the full,  $M$  matrix, the data times are hundreds of times faster. Thus we recommend using the full  $M$  matrix to solve many problems with the same  $A$  and noise characteristics, and the approximate  $M$  when any of  $m$ ,  $n$ ,  $\sigma_1$  or  $\sigma_2$  vary.

Barzilai and Borwein (BB) steps usually improve the accuracy and speed of FPC and GPSR. Thus the timing studies always use the BB-step variants of these algorithms. Furthermore, we use GPSR-BB with monotone line search because the version with non-monotone line search sometimes diverges when  $\bar{\mu}$  exceeds some threshold value, see Figure 5.2. GPSR-BB with monotone line search does not have this problem, and still outperforms the basic algorithm.

Our phase plot results seem to indicate that de-biasing does not expand the  $(\delta, \rho)$  range of solvable compressed sensing problems, but can often improve the accuracy achieved in the recoverable region. The gains are greatest in low noise applications, since in this setting it is much more efficient to stop an algorithm for solving (1.3) short by using a small  $\bar{\mu}$  and de-biasing than to achieve the same accuracy with the optimization algorithm alone.

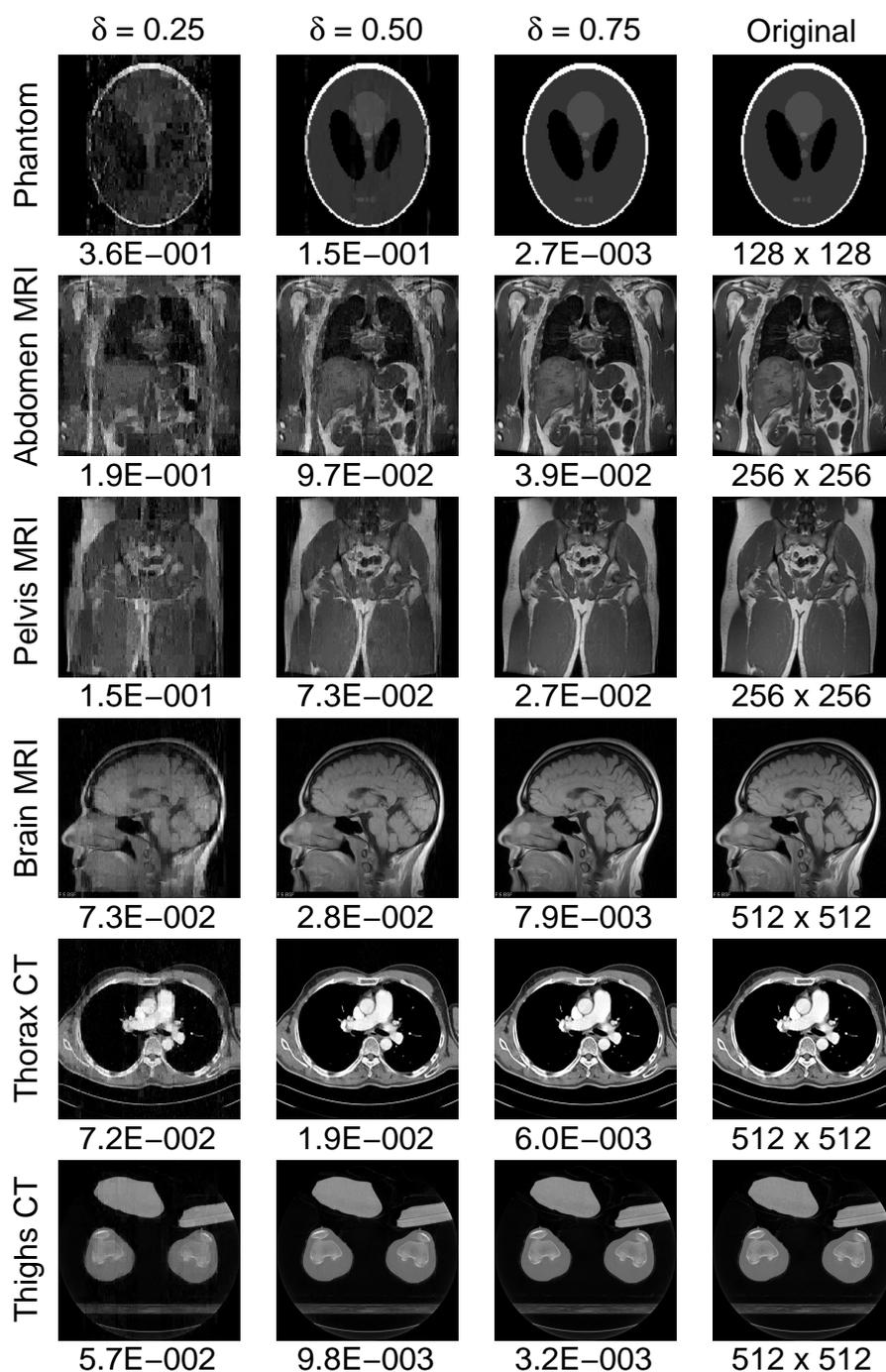


Fig. 4.5. Original and recovered images for Shepp-Logan phantom and five medical images. Each image was recovered using three different measurement:signal length ratios, namely  $\delta = m/n = 0.25, 0.5$  and  $0.75$ . Image size is reported beneath the original image; relative recovery error based on the induced matrix 2-norm is shown below each reconstruction.

StOMP seems to be especially sensitive to signal noise ( $\sigma_1 \neq 0$ ), as its presence, at least in our experiments, resulted in a minimum measurement ratio (approximately  $m/n = 0.1$ ) below

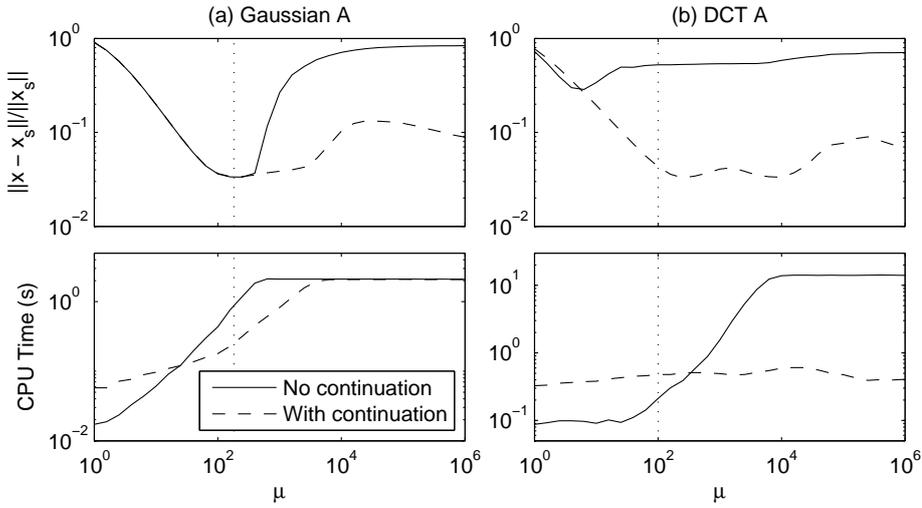


Fig. 5.1. Performance of monotone GPSR-BB without and with continuation, shown as a function of  $\bar{\mu}$ . (a) Problems with Gaussian  $A$ ,  $n = 1024$ ,  $m = 0.3n$ ,  $k = 0.2m$ ,  $\sigma_1 = 1E-2$  and  $\sigma_2 = 1E-8$ ; (b) partial DCT  $A$  matrices,  $n = 8192$ ,  $m = 0.5n$ ,  $k = 0.2m$ ,  $\sigma_1 = 1E-2$  and  $\sigma_2 = 1E-2$ . The vertical dotted lines mark the location of the recommended  $\bar{\mu}$  values (Section 3.1).

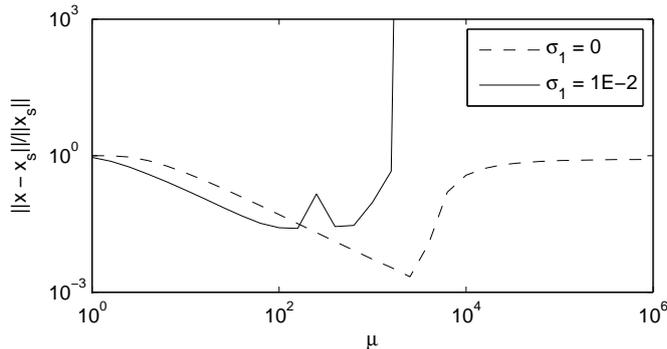


Fig. 5.2. Solution relative error as a function of  $\bar{\mu}$  for the non-monotone variant of GPSR-BB applied to compressed sensing reconstruction problems with Gaussian  $A$ ,  $n = 1024$ ,  $m = 0.3n$ ,  $k = 0.1m$ , and  $\sigma_2 = 1E-8$ . Observe the divergence of the algorithm when  $\sigma_1 = 1E-2$  and  $\bar{\mu}$  exceeds some threshold.

which no signals were recovered even if they were very sparse. Given the limited nature of our experimentation with StOMP’s thresholding parameters, this apparent limitation may be attributable to improper tuning. StOMP is also usually slower than FPC and GPSR, without any outstanding gains in accuracy. It does best with DCT  $A$  matrices and low noise, in which case it sometimes provides the fastest solution.

Of the three algorithms that solve (1.3), `l1_ls` is the most reliable, especially in hard cases when  $\rho = k/m$  is large compared to  $\delta = m/n$ . However, it is typically orders of magnitude slower than FPC and GPSR, with the exception that GPSR and `l1_ls` are about the same speed when noise levels are low and  $\bar{\mu}$  is set to the recommended value. It often achieves the best accuracy, except when the tolerance for its duality gap target is relatively large. Tightening this tolerance should allow `l1_ls` to reach a high accuracy level, as measured by 2-norm relative

error. (As an interior point algorithm,  $l_1$  solutions are dense, and thus do not do well by metrics that focus on correctly identifying zero and nonzero elements.)

The failure of FPC and GPSR to be as accurate as  $l_1$  in hard cases likely has to do with gradient-type method chattering. The problem can be alleviated somewhat by tightening the inner loop stopping tolerances, but it is beyond the scope of this work to determine how to do so automatically (when  $\rho$  is unknown) and without sacrificing speed.

## 6. Conclusions

This paper further develops the algorithm proposed in [40] for Problem (1.1), and performs extensive tests on the resulting algorithms in a series of compressed sensing experiments. Continuation and BB steps are shown to be critical and effective acceleration heuristics, respectively. De-biasing is shown to be useful in low noise situations.

We also provide guidance for choosing the  $\ell_1$  regularization parameter,  $\bar{\mu}$ , and the scaling matrix,  $M$ , based on a measurement model where signal noise and measurement noise can be present at the same time. A novel numerical experiment shows that order of magnitude estimates of noise levels are generally sufficient to obtain good results.

The computational study also compares our fixed-point continuation (FPC) algorithm to three other state-of-the-art algorithms under various noise scenarios with different sparsity levels and problem sizes. The results indicate that FPC is competitive with, and often superior to, the other algorithms, especially when the data are noisy and the problems are large.

Two points of caution should be exercised in evaluating the numerical results in this paper. First, since software tools are still evolving in this fast-moving area, the presented computational results only reflect the performance of the relevant versions of software used in the experiments. Second, most of the problems solved in this work were well within the good region for compressed sensing, that is, relative to the number of non-zero coefficients in the original signal, there were plenty of measurements. In some cases, it may be necessary to push the methodology and use fewer measurements – such situations merit further study.

**Acknowledgements.** The work of E. Hale was supported by an NSF VIGRE grant (DMS-0240058). The work of W. Yin has been supported in part by NSF CAREER Award DMS-0748839 and ONR Grant N00014-08-1-1101. The work of Y. Zhang has been supported in part by NSF Grant DMS-0811188 and ONR Grant N00014-08-1-1101.

## References

- [1] T. Adebeyi and M. Davies, Sparse representations of images using overcomplete complex wavelets, 2005 IEEE/SP 13th Workshop on Statistical Signal Processing, 2005.
- [2] G. Aubert, J. Bect, L. Blanc-Feraud, and A. Chambolle, A  $\ell_1$ -unified variational framework for image restoration, European Conference on Computer Vision, Prague, Lecture Notes in Computer Sciences 3024, IV, 2004.
- [3] D. Baron, M. Wakin, M. Duarte, S. Sarvotham, and R. Baraniuk, Distributed compressed sensing, Preprint, 2005.
- [4] J. Barzilai and J. Borwein, Two point step size gradient methods, *IMA J. Numer. Anal.*, **8** (1988), 141-148.
- [5] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, Massachusetts, second edition, 1999.

- [6] J. Bioucas-Dias, Bayesian wavelet-based image deconvolution: a GEM algorithm exploiting a class of heavy-tailed priors, *IEEE T. Image Process.*, **15**:4 (2006), 937-951.
- [7] J. Bioucas-Dias and M. Figueiredo, A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration, *IEEE T. Image Process.*, **16**:12 (2007), 2992-3004.
- [8] T. Blumensath and M. Davies, Iterative thresholding for sparse approximations, Preprint, 2007.
- [9] T. Blumensath and M. E. Davies, Gradient pursuits, *IEEE T. Signal Proces.*, **56**:6 (2008), 2370-2382.
- [10] S. Bourguignon, H. Carfantan, and J. Idier, A sparsity-based method for the estimation of spectral lines from irregularly sampled data, *IEEE Journal of Selected Topics in Signal Processing*, **1** (2007), 575-585.
- [11] J.-F. Cai, S. Osher, and Z. Shen, Linearized Bregman iterations for frame-based image deblurring, *To appear in SIAM Journal on Imaging Sciences*, 2009.
- [12] E. Candès and J. Romberg, Quantitative robust uncertainty principles and optimally sparse decompositions, *Found. Comput. Math.*, **6**:2 (2006), 227-254.
- [13] E. Candès, J. Romberg, and T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, *IEEE T. Inform. Theory*, **52** (2006), 489-509.
- [14] E. Candès and T. Tao, Near optimal signal recovery from random projections: universal encoding strategies, *IEEE T. Inform. Theory*, **52**:1 (2006), 5406-5425.
- [15] A. Chambolle, An algorithm for total variation minimization and applications, *J. Math. Imaging Vis.*, **20** (2004), 89-97.
- [16] A. Chambolle, R. A. DeVore, N.-Y. Lee, and B. J. Lucier, Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage, *IEEE T. Image Process.*, **7**:3 (1998), 319-335.
- [17] S. Chen, D. Donoho, and M. A. Saunders, Atomic decomposition by basis pursuit, *SIAM J. Sci. Comput.*, **20** (1998), 33-61.
- [18] J. Claerbout and F. Muir, Robust modelling of erratic data, *Geophysics*, **38** (1973), 826-844.
- [19] P. L. Combettes and J.-C. Pesquet, Proximal thresholding algorithm for minimization over orthonormal bases, *To appear in SIAM J. Optimiz.*, 2007.
- [20] Y. Dai and R. Fletcher, Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming, *Numer. Math.*, **100** (2005), 21-47.
- [21] J. Darbon and S. Osher, Fast discrete approximations for sparse approximation and deconvolution, Technical Report (to appear), UCLA, 2007.
- [22] I. Daubechies, M. Defrise, and C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Commun. Pur. Appl. Math.*, **57** (2004), 1413-1457
- [23] I. Daubechies, M. Fornasier, and I. Loris, Accelerated projected gradient method for linear inverse problems with sparsity constraints, *arXiv:0706:4297*, 2007.
- [24] C. De Mol and M. Defrise, A note on wavelet-based inversion algorithms, *Contemporary Mathematics*, **313** (2002), 85-96.
- [25] D. Donoho, Compressed sensing, *IEEE T. Inform. Theory*, **52** (2006), 1289-1306.
- [26] D. Donoho and J. Tanner, Neighborliness of randomly-projected simplices in high dimensions, *Proc. National Academy of Sciences*, **102**:27 (2005), 9452-9457.
- [27] D. Donoho, Y. Tsaig, I. Drori, and J.-C. Starck, Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit, *Submitted to IEEE T. Inform. Theory*, 2006.
- [28] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, Least angle regression, *Ann. Stat.*, **32** (2004), 407-499.
- [29] M. Elad, Why simple shrinkage is still relevant for redundant representations? *IEEE T. Inform. Theory*, **52**:12 (2006), 5559-5569.
- [30] M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky, A wide-angle view at iterated shrinkage algorithms, *SPIE (Wavelet XII), San-Diego CA, August 26-29, 2007.*, 2007.

- [31] M. Elad, B. Matalon, and M. Zibulevsky, Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization, *Journal on Applied and Computational Harmonic Analysis*, 2006.
- [32] M. J. Fadili and J. L. Starck, Sparse representation-based image deconvolution by iterative thresholding, *Astronomical Data Analysis ADA'06*, Marseille, France, 2006.
- [33] M. Figueiredo, J. Bioucas-Dias, and R. Nowak, Majorization-minimization algorithms for wavelet-based image restoration, *IEEE T. Image Process.*, **16**:12 (2007), 2980–2991.
- [34] M. Figueiredo and R. Nowak, An EM algorithm for wavelet-based image restoration. *IEEE T. Image Process.*, **12** (2003), 906-916.
- [35] M. Figueiredo and R. Nowak, A bound optimization approach to wavelet-based image deconvolution, *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2005.
- [36] M. Figueiredo, R. Nowak, and S. J. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, To appear in the *IEEE Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing*, 2007.
- [37] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, Pathwise coordinate optimization, *Annals of Applied Statistics*, **1** (2007), 302-332.
- [38] T. Goldstein and S. Osher, The split Bregman algorithm for L1 regularized problems, *UCLA CAM Report 08-29*, 2008.
- [39] E. Hale, W. Yin, and Y. Zhang, A fixed-point continuation method for  $\ell_1$ -regularization with application to compressed sensing, *Rice University CAAM Technical Report TR07-07*, 2007.
- [40] E. T. Hale, W. Yin, and Y. Zhang, Fixed-point continuation for  $\ell_1$ -minimization: Methodology and convergence, *SIAM J. Optimiz.*, **19** (2008), 1107-1130.
- [41] E. T. Hale, W. Yin, and Y. Zhang, A numerical study of fixed-point continuation applied to compressed sensing, *Technical Report TR08-24*, Rice University, Department of Computational and Applied Mathematics, 2008.
- [42] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, A method for large-scale  $\ell_1$ -regularized least squares problems with applications in signal processing and statistics, [http://www.stanford.edu/~boyd/11\\_ls.html](http://www.stanford.edu/~boyd/11_ls.html). 2007.
- [43] S. Kirolos, J. Laska, M. Wakin, M. Duarte, D. Baron, T. Ragheb, Y. Massoud, and R. Baraniuk, Analog-to-information conversion via random demodulation, In *Proceedings of the IEEE Dallas Circuits and Systems Workshop (DCAS)*, Dallas, Texas, 2006.
- [44] J. Laska, S. Kirolos, M. Duarte, T. Ragheb, R. Baraniuk, and Y. Massoud, Theory and implementation of an analog-to information converter using random demodulation, In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, New Orleans, Louisiana, 2007.
- [45] J. Laska, S. Kirolos, Y. Massoud, R. Baraniuk, A. Gilbert, M. Iwen, and M. Strauss, Random sampling for analog-to-information conversion of wideband signals, In *Proceedings of the IEEE Dallas Circuits and Systems Workshop*, Dallas, Texas, 2006.
- [46] S. Levy and P. Fullagar, Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution, *Geophysics*, **46** (1981), 1235-1243.
- [47] Z.-Q. Luo and P. Tseng, On the linear convergence of descent methods for convex essentially smooth minimization, *SIAM J. Control Optimiz.*, **30**:2 (1990), 408-425.
- [48] M. Lustig, D. Donoho, and J. Pauly, Sparse MRI: The application of compressed sensing for rapid MR imaging, *Preprint*, 2007.
- [49] S. Ma, W. Yin, Y. Zhang, and A. Chakraborty, An efficient algorithm for compressed MR imaging using total variation and wavelets, *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2008*, (1–8), 2008.
- [50] S. G. Mallat and Z. Zhang, Matching pursuits with time-frequency dictionaries, *IEEE T. Signal Process.*, **41**:12 (1993), 3397-3415.

- [51] A. Miller, Subset Selection in Regression, Chapman and Hall, 2002.
- [52] Y. Nesterov, Gradient methods for minimizing composite objective function, CORE Discussion Paper 2007/76, 2007. <http://www.caam.rice.edu/~optimization/L1/fpc/>.
- [53] R. Nowak and M. Figueiredo, Fast wavelet-based image deconvolution using the EM algorithm, Proceedings of the 35th Asilomar Conference on Signals, Systems, and Computers, Monterey, CA, 2001.
- [54] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, An iterated regularization method for total variation based image restoration, *SIAM Journal on Multiscale Modeling and Simulation*, **4:2** (2005), 460-489.
- [55] S. Osher, Y. Mao, B. Dong, and W. Yin, Fast linearized Bregman iteration for compressive sensing and sparse denoising, Rice University CAAM Technical Report TR08-07, 2008.
- [56] C. C. Paige and M. A. Saunders, Lsqr: An algorithm for sparse linear equations and sparse least squares, *ACM T. Math. Softmart.*, **8:1** (1982), 43-71.
- [57] J.-S. Pang, A posteriori error bounds for the linearly-constrained variational inequality problem, *Math. Method. Oper. Res.*, **12** (1987), 474-484.
- [58] T. H. Reeves and N. G. Kingsbury, Overcomplete image coding using iterative projection-based noise shaping, 2002 International Conference on Image Processing, **3** (2002), 24-28.
- [59] M. Rudelson and R. Vershynin, Geometric approach to error correcting codes and reconstruction of signals, *International Mathematical Research Notices*, **64** (2005), 4019-4041.
- [60] F. Santosa and W. Symes, Linear inversion of band-limited reflection histograms, *SIAM J. Sci. Comput.*, **7** (1986), 1307-1330.
- [61] J. Shi, W. Yin, S. Osher, and P. Sajda, An algorithm for large-scale  $\ell_1$ -regularized logistic regression, In preparation, 2008.
- [62] D. Takhar, J. Laska, M. Wakin, M. Duarte, D. Baron, S. Sarvotham, K. Kelly, and R. Baraniuk, A new compressive imaging camera architecture using optical-domain compression, In Proceedings of Computational Imaging IV at SPIE Electronic Image, San Jose, California, 2006.
- [63] H. Taylor, S. Bank, and J. McCoy, Deconvolution with the  $l_1$  norm, *Geophysics*, **44** (1979), 39-52.
- [64] R. Tibshirani, Regression shrinkage and selection via the lasso, *J. Roy. Stat. Soc. B*, **58** (1996), 267-288.
- [65] J. Tropp, Just relax: Convex programming methods for identifying sparse signals, *IEEE T. Inform. Theory*, **51** (2006), 1030-1051.
- [66] J. Tropp and A. Gilbert, Signal recovery from partial information via orthogonal matching pursuit, *IEEE T. Inform. Theory*, **53:12** (2007), 4655-4666.
- [67] J. Tropp, M. Wakin, M. Duarte, D. Baron, and R. Baraniuk, Random filters for compressive sampling and reconstruction, In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toulouse, France, 2006.
- [68] Y. Tsaig and D. Donoho, Extensions of compressed sensing, *Signal Process.*, **86:3** (2005), 533-548.
- [69] P. Tseng, Convergence of block coordinate descent method for nondifferentiable minimization, *J. Optimiz. Theory App.*, **109:3** (2001), 1573-2878.
- [70] P. Tseng and S. Yun, A coordinate gradient descent method for nonsmooth separable minimization, *Math. Program.*, **117:1-2** (2009), 387-423.
- [71] E. Van den Berg and M. P. Friedlander, SPGL1: A MATLAB solver for large-scale sparse reconstruction, <http://www.cs.ubc.ca/labs/scl/index.php/main/spgl1>. 2007.
- [72] E. van den Berg, M. P. Friedlander, G. Hennenfent, F. J. Herrmann, Y. Saad, and P. Yilmaz, Sparco: A testing framework for sparse reconstruction. UBC Computer Science Technical Report TR-2007-20, 2007.
- [73] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, An architecture for compressing image, In Proceedings of the International Conference on Image Processing (ICIP), Atlanta, Georgia, 2006.
- [74] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk,

- Compressive imaging for video representation and coding, In Proceedings of Picture Coding Symposium (PCS), Beijing, China, 2006.
- [75] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization and continuation, *Submitted to SIAM Journal on Scientific Computing, Rice University CAAM Technical Report TR09-01*, 2009.
  - [76] S. Wright, R. Nowak, and M. Figueiredo, Sparse reconstruction by separable approximation, Submitted, 2008.
  - [77] J. Yang, Y. Zhang, and W. Yin, A fast TVL1-L2 algorithm for image reconstruction from partial fourier data, Submitted to IEEE Journal of Selected Topics in Signal Processing Special Issue on Compressed Sensing. Rice University CAAM Technical Report TR08-27, 2008.
  - [78] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, Bregman iterative algorithms for  $\ell_1$ -minimization with applications to compressed sensing, *SIAM Journal on Imaging Sciences*, **1:1** (2008), 143-168.
  - [79] S. Yun and K.-C. Toh, A coordinate gradient descent method for L1-regularized convex minimization, *Optimization-Online*, 2008.
  - [80] H. Zhang and W. W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, *SIAM Journal on Optimization*, **14:4** (2004), 1043-1056.
  - [81] Y. Zhang, When is missing data recoverable? Rice University CAAM Technical Report TR06-15, 2006.