CAAM Technical Report TR07-07

# A Fixed-Point Continuation Method for $\ell_1$-Regularized Minimization with Applications to Compressed Sensing

Elaine T. Hale, Wotao Yin, and Yin Zhang

Department of Computational and Applied Mathematics
Rice University, Houston, Texas, 77005, U.S.A.

July 7, 2007

### Abstract

We consider solving minimization problems with $\ell_1$-regularization:

$$\min \|x\|_1 + \mu f(x),$$

particularly for $f(x) = \frac{1}{2}\|Ax - b\|_M^2$, where $A \in \mathbb{R}^{m\times n}$ and $m < n$. Our goal is to construct efficient and robust algorithms for solving large-scale problems with dense data, and our approach is based on two powerful algorithmic ideas: operator-splitting and continuation. This paper establishes, among other results, $q$-linear convergence rates for our algorithm applied to problems with $f(x)$ convex, but not necessarily strictly convex. We present numerical results for several types of compressed sensing problems, and show that our algorithm compares favorably with several state-of-the-art algorithms when applied to large-scale problems with noisy data.

**Keywords:** $\ell_1$ regularization, fixed point algorithm, $q$-linear convergence, continuation, compressed sensing.

# Contents

# 1 Introduction

Under suitable conditions, minimizing the $\ell_1$-norm is equivalent to minimizing the so-called "$\ell_0$-norm", that is, the number of nonzeros in a vector. The former is always computationally more tractable than the latter. Thus, minimizing or limiting the magnitude of $\|x\|_1$ has long been recognized as a practical avenue for obtaining sparse solutions $x$. Some early work is in the area of geophysics, where sparse spike train signals are often of interest, and data may include large sparse errors [8, 31, 46, 48]. The signal processing and statistics communities use the $\ell_1$-norm to describe a signal with just a few waveforms or a response variable with just a few explanatory variables, respectively [36, 7, 49, 19]. More references on $\ell_1$-regularization for signal processing and statistics can be found in [38].

As a general principle, a sparse solution $x \in \mathbb{R}^n$ to an under-determined linear system of equations $Ax = b$ may be obtained by minimizing or constraining the $\ell_1$-norm of $x$. If the "observation" $b$ is contaminated with noise, then an appropriate norm of the residual $Ax - b$ should be minimized or constrained (which norm to use depends on the nature of the noise). Such considerations yield a family of related optimization problems. If there is no noise in $b$, but the solution $x$ should be sparse, then it is appropriate to solve the basis pursuit problem [7]

$$\min_{x\in\mathbb{R}^n} \left\{\|x\|_1 \,|\, Ax = b\right\}. \tag{1}$$

However, if there is Gaussian noise in $b$, the $\ell_1$-regularized least squares problem

$$\min_{x\in\mathbb{R}^n} \|x\|_1 + \frac{\mu}{2}\|Ax - b\|_2^2, \tag{2}$$

would be appropriate, as would the Lasso problem [49]

$$\min_{x\in\mathbb{R}^n} \left\{\|Ax - b\|_2^2 \,|\|x\|_1 \leq t\right\}, \tag{3}$$

which is equivalent to (2) given appropriate constants $\mu$ and $t$. Similarly, other formulations of interest may be generated by choosing appropriate norms and variously specifying that the resulting terms be minimized or constrained.

## 1.1 $\ell_1$-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 minimizing the $\ell_1$-norm (or its variants) in order to decode the signal. New results reported by Candes *et al* [4, 2, 3], Donoho *et al* [12, 53, 13] and others [45, 51] stimulated the current burst of research in this area. Applications of compressed sensing include compressive imaging [56, 57, 47], medical imaging [34], multi-sensor and distributed compressed sensing [1], analog-to-information conversion [52, 27, 30, 29], and missing data recovery [58]. Compressed sensing is attractive for these and other potential applications because it reduces the number of measurements required to obtain a given amount of information. The trade-off is the addition of a non-trivial decoding process.

In brief, compressed sensing theory shows that a sparse signal of length $n$ can be recovered from $m < n$ measurements by solving any appropriate variant of (1), (2), (3), 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 orthonormal matrices (such as partial discrete cosine transform (DCT) matrices) have been proven to be "good". These matrices are invariably dense, which contradicts the usual assumption made by conventional optimization solvers that large-scale problems appear with sparse data. Thus it is necessary to develop dedicated algorithms for compressed sensing signal reconstruction. Further, the size and density of the data involved suggest that those algorithms should not require large linear system solves or matrix factorizations, and should take full advantage of available fast transforms like FFT and DCT.

In this work, we present an algorithmic framework and related convergence analysis for solving general problems of the form:

$$\min_{x\in\mathbb{R}^n} \|x\|_1 + \mu f(x), \tag{4}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable and convex, but not necessarily strictly convex, and $\mu > 0$. Our numerical studies, on the other hand, are focused on the following generalization of (2):

$$\min_{x\in\mathbb{R}^n} \|x\|_1 + \frac{\mu}{2}\|Ax - b\|_M^2, \tag{5}$$

where $M \in \mathbb{R}^{m\times m}$ is a positive definite matrix, $\|x\|_M := \sqrt{x^\top Mx}$ is the associated $M$-norm, $A \in \mathbb{R}^{m\times n}$ is dense, $m \leq n$ or even $m \ll n$, and $n$ is large. We emphasize that our algorithm does not require any linear system solves or matrix factorizations—we restrict ourselves to vector operations and matrix-vector multiplications.

## 1.2 Our Approach and Main Results

The objective function in (4) is the sum of two convex functions. While the $\ell_1$-norm term is not smooth, it is easily transformed into a linear function plus some linear constraints, such that standard interior-point methods utilizing a direct linear solver can be applied to our problem of interest, (5). However, this standard approach is too costly for large-scale problems with dense data. Alternative algorithms include utilizing an iterative linear solver in an interior-point framework [26], and using a gradient projection method [23].

Our approach is based on operator splitting. It is well known in convex analysis that minimizing a convex function $\phi(x)$ is equivalent to finding a zero of the subdifferential $\partial\phi(x)$, i.e., finding $x$ such that $\mathbf{0} \in \partial\phi(x) := T(x)$, where $T$ is a maximal monotone operator [44]. In many cases, one can split $\phi$ into a sum of two convex functions, $\phi = \phi_1 + \phi_2$, which implies the decomposition of $T$ into the sum of two maximal monotone operators $T_1$ and $T_2$, i.e., $T = T_1 + T_2$. For some $\tau > 0$, if $T_2$ is single-valued and $(I + \tau T_1)$ is invertible, then

$$\begin{aligned} \mathbf{0} \in T(x) \quad &\Longleftrightarrow \quad \mathbf{0} \in (x + \tau T_1(x)) - (x - \tau T_2(x)) \\ &\Longleftrightarrow \quad (I - \tau T_2)x \in (I + \tau T_1)x \\ &\Longleftrightarrow \quad x = (I + \tau T_1)^{-1}(I - \tau T_2)x. \end{aligned} \tag{6}$$

Equation (6) leads to the *forward-backward splitting* algorithm for finding a zero of $T$:

$$x^{k+1} := (I + \tau T_1)^{-1}(I - \tau T_2)x^k, \tag{7}$$

which is a fixed point algorithm. For the minimization problem (4), $T_2 = \mu\nabla f$ and $(I + \tau T_1)^{-1}$ is component-wise shrinkage (or soft shresholding), the details of which are given in Sections 3 and 4 below.

The forward-backward splitting method was first proposed by Lions and Mercier [32], and Passty [42] around the same time in 1979. Over the years, this scheme and its modifications have been extensively studied by various authors including, to name a few, Mercier [37], Gabay [24], Eckstein [17], Chen and Rockafellar [6], Noor [39], and Tseng [54]. However, the idea of splitting operators can be traced back to the works of Peaceman and Rachford [43] and Douglas and Rachford [16] in the mid 1950's for solving second-order elliptic and parabolic partial differential equations.

General convergence theory has been developed for forward-backward splitting methods [37, 24, 6]. Unfortunately, the existing theory requires rather strong conditions on $T_2$, or on $T$ as a whole. In short, when reduced to our setting, with $\phi_1 = \|x\|_1$ and $\phi_2 = \mu f(x)$, the existing convergence theory requires either $f$ or the whole objective in (4) to be strictly convex. Stronger convergence results generally require modifications to the basic algorithm [54].

Daubechies *et al* [10] (see also [20, 9]) recently proposed the application of the fixed point algorithm (7) to the problem (2). In [10], the authors derived the fixed point scheme through a surrogate function, apparently unaware of the connection to operator splitting. They analyzed the quadratic model (2), as opposed to the more general model (4), and proved global convergence for a specific choice of $\tau$ without any strict convexity requirement, but did not analyze the rate of convergence. The authors concluded in [10] that when applied to large-scale problems with dense data, the cost of the fixed point algorithm (7) "may be too heavy, computationally, to compete effectively with, e.g., modern high-dimensional optimization methods."

In the present work, we aim to addresses the following two questions. Can strong convergence results, especially on rate of convergence, be obtained for algorithm (7) applied to the general problem (4)? Can algorithm (7) be made computationally competitive for problem (5)? Our answers to both questions are affirmative.

On the theoretical side, we have obtained global convergence, even finite convergence for some quantities, and a $q$-linear rate of convergence without assuming strict convexity, nor uniqueness of solution. Furthermore, we show that the rate of convergence is not determined by the conditioning of the Hessian of $f$, as normally is the case for gradient-type methods, but by that of a reduced Hessian whose condition number can be much smaller than that of the Hessian when $A$ has many fewer rows than columns.

On the computational side, we have devised a continuation strategy that significantly reduces the number of iterations required for a given value of $\mu$. Our numerical results indicate that our algorithm is especially suited for large-scale instances of problem (2) when $A$ is a partial transform matrix such as a DCT matrix. In comparison with several recently developed algorithms, our algorithm is clearly most robust, and in many cases also the fastest.

### 1.3 Several Recent Algorithms

Orthogonal Matching Pursuit (OMP) [36, 50] and its successor Stagewise Orthogonal Matching Pursuit (StOMP) [15] do not solve (5) per se, but use a greedy approach to identify which columns of $A$ were used to construct the linear combination $b = Ax_s$, where $x_s$ is the original sparse signal. Both rely on the residual vector $r = b - B\hat{x}$, where $B$ is a matrix whose columns are a subset of the columns of $A$, and $\hat{x}$ is the solution to the least squares problem $\min_x \|Bx - b\|_2^2$. The algorithms identify columns of $A$ to add to $B$ by examining the correlations $A^T r$. The primary computational costs of OMP and StOMP are matrix-vector multiplications and the solution of the least squares problems. Since the size of the least squares problems depends primarily on the number of columns of $B$, these algorithms should be fast for very sparse signals. Conceptually, StOMP represents a variant of OMP in that more than one column may be added to $B$ during a given iteration, so that fewer iterations are required. Meanwhile, StOMP has a reduced probability of solving (5) exactly, and it is not trivial to specify an appropriate thresholding scheme and the associated tuning parameter when applying StOMP to novel $A$ matrices, or in the presence of noise.

There are some algorithms for solving (2) whose main computational costs are matrix-vector multiplications. Gradient Projection for Sparse Reconstruction (GPSR) was recently proposed by Figueiredo, Nowak and Wright [23]. They reformulate (2) as a bound-constrained quadratic program, and then apply projected gradient steps, optionally using a variant of Barzilai-Borwein steps (GPSR-BB) in order to accelerate convergence. We mention that despite the success of the BB steps for unconstrained optimization, not much is known about its convergence behavior for constrained optimization.

The algorithm of Kim, Koh, Lustig and Boyd [26] was developed for a large-scale least squares problem with $\ell_1$ regularization that is slightly different than, but completely equivalent to (2). The authors employ a specialized interior-point method that uses a preconditioned conjugate gradient (PCG) method to approximately solve linear systems in a truncated-Newton framework. The authors exploit the structure of the involved Hessian to construct their preconditioner and show that accurate solutions for Magnetic Resonance Imaging (MRI) problems can be obtain with around a hundred PCG steps. The Matlab code for their algorithm is named $l1\_ls$.

Some earlier algorithms that either utilize PCG or just use matrix-vector multiplications include [7, 28, 21, 22]. However, numerical results reported in [15, 23, 26] indicate that these earlier algorithms are not as efficient as the more recent ones mentioned above. In Section 7, we will compare our algorithm to the three recent algorithms: StOMP, GPSR and $l1\_ls$.

### 1.4 Notation and Organization

For simplicity, we let $\|\cdot\| := \|\cdot\|_2$, the Euclidean norm, unless otherwise specified. The *support* of $x \in \mathbb{R}^n$ is $\text{supp}(x) := \{i : x_i \neq 0\}$. Let

$$g(x) := \nabla f(x)$$

be the gradient of $f(x)$; in particular, $g(x) = A^\top M(Ax - b)$ for $f$ defined by (5), that is

$$f = \frac{1}{2}\|Ax - b\|_M^2. \tag{8}$$

For a set $E$, we use $|E|$ to denote that its cardinality. For any matrix $M \in \mathbb{R}^{n\times n}$, we denote its eigenvalues as $\lambda_i(M)$, $i = 1, \ldots, n$, and its maximum and minimum eigenvalues as, respectively, $\lambda_{\max}(M)$ and $\lambda_{\min}(M)$.

The signum function of $t \in \mathbb{R}$ is

$$\text{sgn}(t) := \begin{cases} +1 & t > 0, \\ 0 & t = 0, \\ -1 & t < 0; \end{cases}$$

while the signum multifunction (i.e., set-valued function) 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 $|t|$. For $x \in \mathbb{R}^n$, we define $\mathrm{sgn}(x) \in \mathbb{R}^n$ and $\mathrm{SGN}(x) \subset \mathbb{R}^n$ component-wise as $(\mathrm{sgn}(x))_i := \mathrm{sgn}(x_i)$ and $(\mathrm{SGN}(x))_i := \mathrm{SGN}(x_i)$, $i = 1, 2, \cdots, n$, respectively. Clearly,

$$\mathrm{sgn}(x) = \mathrm{sgn}(x') \iff \mathrm{SGN}(x) = \mathrm{SGN}(x'),\ \forall\ x, x'.$$

For $x, y \in \mathbb{R}^n$, let $x \odot y \in \mathbb{R}^n$ denote the component-wise product of $x$ and $y$, i.e., $(x \odot y)_i = x_i y_i$. Furthermore, vector operators such as $|x|$ and $\max\{x, y\}$ are defined to operate component-wise, analogous with the definitions of sgn and SGN.

For any index set $I \subseteq \{1, 2, \ldots, n\}$ (later, we will use index sets $E$ and $L$), $x_I$ is defined as the sub-vector of $x$ of length $|I|$ consisting only of components $x_i$, $i \in I$. Similarly, if $g$ is a vector-valued function, then $g_I(x)$ denotes the sub-vector of $g(x)$ consisting of $g_i(x)$, $i \in I$.

This paper is organized as follows. In Section 2, we recall the classic optimality (or in general, stationarity) conditions for problem (4), and then characterize the optimal solution sets of problems (4) and (5). In Section 3, we present a fixed point optimality condition for (4). This optimality condition motivates a fixed point algorithm and introduces a shrinkage operator, the properties of which conclude Section 3. In Section 4, we present our results on the convergence and rate of convergence of the fixed point algorithm; the proofs of the main results are given in Section 5. We motivate and propose a continuation method in Section 6 and briefly discuss a number of possible extensions. Section 7 contains the details of our implementation of the fixed point algorithm as applied to (5) and numerical results comparing our method to others. Finally, we conclude the paper in Section 8.

## 2 Optimality and Optimal Solution Set

Recall that $f(x)$ in (4) is convex and let $X^*$ be the set of optimal solutions of (4). It is well-known from convex analysis (see, for example, [44]) that an optimality condition for (4) is

$$x^* \in X^* \iff \mathbf{0} \in \mathrm{SGN}(x^*) + \mu g(x^*), \tag{9}$$

where $\mathbf{0}$ is the zero vector in $\mathbb{R}^n$; or equivalently,

$$x^* \in X^* \iff \mu g_i(x^*) \begin{cases} = -1, & x_i^* > 0, \\ \in [-1, 1], & x_i^* = 0, \\ = 1, & x_i^* < 0. \end{cases} \tag{10}$$

It follows readily from (10) that $\mathbf{0}$ is an optimal solution of (4) if and only if $\mu \|g(\mathbf{0})\|_\infty \le 1$, or in other words,

$$\mathbf{0} \in X^* \iff \mu \le \frac{1}{\|g(\mathbf{0})\|_\infty}. \tag{11}$$

Therefore, it is easy to check whether $\mathbf{0}$ is a solution of (4).

The following theorem establishes some properties of $X^*$ that are of interest in their own right, but will also be useful in later developments.

**Theorem 2.1.** *Let $f \in C^2$ be convex and $X^*$ be the set of optimal solutions of* (4)*, which is nonempty.*

*i. If $x^1 \in X^*$ and $x^2 \in X^*$, then $g(x^1) = g(x^2)$.*

*ii.* *$x^* \in X^*$ if and only if $g(x^*) \equiv g^*$, where for $i = 1, 2, \ldots, n$,*

$$\mu g_i^* \begin{cases} = -1, & \max\{x_i : x \in X^*\} > 0, \\ = +1, & \min\{x_i : x \in X^*\} < 0, \\ \in [-1, 1], & \textit{otherwise}. \end{cases} \tag{12}$$

*iii.* *$X^*$ is contained in a single orthant of $\mathbb{R}^n$, namely*

$$X^* \subset O := \{x \in \mathbb{R}^n : -\mathrm{sgn}^+(g_i^*)x_i \geq 0, \forall i\}, \tag{13}$$

*where $\mathrm{sgn}^+(\cdot)$ is equal to $\mathrm{sgn}(\cdot)$ except $\mathrm{sgn}^+(0) = 1$, i.e.,*

$$\mathrm{sgn}^+(t) := \begin{cases} +1 & t \geq 0, \\ -1 & t < 0. \end{cases}$$

*(In addition, we let $\mathrm{sgn}^+(x)$ be defined component-wise for any $x \in \mathbb{R}^n$.)*

*Furthermore, if $f(x)$ is defined as in (8), then*

*iv.* *If $x^1 \in X^*$ and $x^2 \in X^*$, then $Ax^1 = Ax^2$.*

*v.* *$\|x^*\|_1$ and $\|Ax^* - b\|_M$ are constant for all $x^* \in X^*$.*

*vi.* *$X^*$ is a bounded polyhedron, i.e., a polytope.*

*Proof.* We prove the statements one by one.

i. This part will be proven later as Corollary 4.1 under Assumption 1, which is slightly weaker than what is assumed for this theorem. That proof is independent of this theorem and the results that follow from it.

ii. (12) follows directly from i. and (10) applied to all $x^* \in X^*$.

iii. From (9) and (12), if there exists an $x^* \in X^*$ with a strictly positive (negative) $x_i^*$, then $\mu g_i^* = -1$ ($\mu g_i^* = 1$), so all other $x \in X^*$ must satisfy $x_i \geq 0$ ($x_i \leq 0$). Consequently, $X^*$ lies in the orthant $O$.

iv. From i. and for $f(x)$ so specified, $g(x^1) - g(x^2) = A^\top MA(x^1 - x^2) = \mathbf{0}$, which immediately implies that $A(x^1 - x^2) = \mathbf{0}$, given that $M$ is symmetric positive definite.

v. From iv., $Ax^*$ is constant over $X^*$, and so $\|Ax^* - b\|_M$ must be as well. Since (5) has a unique optimal objective value, $\|x^*\|_1$ must also be constant.

vi. Defining $p = -\mathrm{sgn}^+(g^*)$, from the definition of $O$ we have

$$p^\top x = \|x\|_1, \ \forall x \in O.$$

Consider the linear program

$$\min_x \{p^\top x : Ax = c, \ x \in O\} \tag{14}$$

where $c = Ax^*$ for any $x^* \in X^*$. It is easy to verify that an optimal solution $\bar{x}$ of (14) satisfies both $\|\bar{x}\|_1 = \|x^*\|_1$ and $\|A\bar{x} - b\|_M = \|Ax^* - b\|_M$ for any $x^* \in X^*$ and vice versa. So (14) is equivalent to (2), as long as $c$ and $O$ (or equivalently, $g^*$) are known. Consequently, $X^*$, as the solution set of the linear program (14), must be a polyhedron and must be bounded since $\|x^*\|_1$ is constant for all $x^* \in X^*$.

This completes the proof. □

# 3 A Fixed-Point Algorithm

## 3.1 Optimality as a Fixed-Point Equation

We start by stating another optimality condition for problem (4). Although this condition can be derived from an operator-splitting argument, here we give a simple and direct proof.

**Proposition 3.1.** *Recall that $X^*$ is the set of optimal solutions of* (4)*. For any scalar $\tau > 0$, $x^* \in X^*$ if and only if*

$$x^* = \operatorname{sgn}(x^* - \tau g(x^*)) \odot \max\left\{|x^* - \tau g(x^*)| - \frac{\tau}{\mu}, \mathbf{0}\right\}. \tag{15}$$

*Proof.* Assume that $x$ satisfies equation (15). When $x_i > 0$, we have $\operatorname{sgn}(x_i - \tau g_i(x)) = 1$ and $x_i - \tau g_i(x) - \tau/\mu > 0$, from which it follows that

$$\begin{aligned} x_i &= \operatorname{sgn}(x_i - \tau g_i(x)) \max\{|x_i - \tau g_i(x)| - \tau/\mu, 0\} \qquad (16)\\ &= x_i - \tau g_i(x) - \tau/\mu, \end{aligned}$$

and $\mu g_i(x) = -1$. On the other hand, when $x_i > 0$, $\mu g_i(x) = -1$ also gives (16).

Similarly, (16) is equivalent to $\mu g_i(x) = 1$ when $x_i < 0$.

When $x_i = 0$, $x_i$ satisfies (16) if and only if

$$|x_i - \tau g_i(x)| = |\tau g_i(x)| = \tau |g_i(x)| \leq \tau/\mu,$$

which is equivalent to $\mu g_i(x) \in [-1, 1]$.

Since we considered all cases for the sign of $x_i$, (15) is equivalent to (10) and thus (9). □

**Remark.** Proposition 3.1 still holds if the strictly positive scalar $\tau$ is replaced by any mapping $d : \mathbb{R}^n \to \{x \in \mathbb{R}^n : x > \mathbf{0}\}$, and (15) by

$$x^* = \operatorname{sgn}(x^* - d(x^*) \odot g(x^*)) \odot \max\left\{|x^* - d(x^*) \odot g(x^*)| - \frac{d(x^*)}{\mu}, \mathbf{0}\right\}. \tag{17}$$

The proof of Proposition 3.1 will carry over to this more general result as it is based on the component-wise analysis. The algorithm and analysis below based on (15) can also be extended in a similar way with little effort.

The right-hand side of the fixed point equation (15) is a composition of two mappings $s_\nu$ and $h$ from $\mathbb{R}^n$ to $\mathbb{R}^n$ defined as:

$$h(\cdot) \quad := \quad I(\cdot) - \tau g(\cdot), \tag{18}$$

$$s_\nu(\cdot) \quad := \quad \operatorname{sgn}(\cdot) \odot \max\{|\cdot| - \nu, 0\}, \text{ where } \nu > 0. \tag{19}$$

Intuitively, $h(\cdot)$ resembles a gradient descent step (for decreasing $f(x)$), and $s_\nu(\cdot)$ reduces the magnitude of each nonzero component of the input vector by an amount less than or equal to $\nu$, thus reducing the $\ell_1$-norm. Later we will also use $s_\nu$ as a mapping from $\mathbb{R}$ to $\mathbb{R}$ in composition with which $h_i(\cdot) = (h(\cdot))_i$ from $\mathbb{R}^n$ to $\mathbb{R}$.

For solving the equation (15) and thus the minimization problem (4), it is natural to consider *the fixed point iterations*:

$$x^{k+1} = s_\nu \circ h(x^k), \tag{20}$$

for $\nu = \tau/\mu$.

Although we originally derived the fixed point scheme (20) from a totally different approach, we later found that it is just the forward-backward splitting algorithm (7) with

$$T_1(x) = \partial\|x\|_1/\mu, \quad T_2(x) = g(x).$$

Through simple calculations we can derive

$$s_\nu = (I + \tau T_1)^{-1}, \quad h = I - \tau T_2.$$

However, some special properties of the operator $s_\nu$, given below, allow us to obtain strong convergence results that cannot be directly inferred from the existing theory for forward-backward splitting algorithms applied to more general operators.

### 3.2 Properties of Shrinkage Operator

It is easy to verify that $s_\nu(y)$ is the unique solution of

$$\min_{x\in\mathbb{R}^n} \nu\|x\|_1 + \frac{1}{2}\|x-y\|^2$$

for any $y \in \mathbb{R}^n$. Wavelet analysts refer to $s_\nu(\cdot)$ as the soft-thresholding [11] or wavelet shrinkage (see [5], for example) operator. For convenience, we will refer to $s_\nu(\cdot)$ as the *shrinkage operator*.

The two lemmas below establish some useful properties of the shrinkage operator. Both make immediate use of the component-wise separability of $s_\nu$, that is, for all indices $i$

$$(s_\nu(y))_i = s_\nu(y_i).$$

The alternative representation of $s_\nu$ in Lemma 3.1 will be used to prove Lemma 3.2. Lemma 3.2 proves a number of component-wise properties of $s_\nu$, including non-expansiveness. These results will be used in the component-wise convergence analysis of the iteration sequence $\{x^k\}$ in Section 4.

**Lemma 3.1.** *Let $[x]^+ := \max\{x, \mathbf{0}\}$ denote the orthogonal projection of a vector $x \in \mathbb{R}^n$ onto the first orthant. We have*

$$s_\nu(y) = y - \nu\mathrm{sgn}(y) + \mathrm{sgn}(y) \odot [\nu - |y|]^+, \text{ for any } y \in \mathbb{R}^n. \tag{21}$$

*Furthermore, if $\nu - |y_i| > 0$, then $(s_\nu(y))_i = s_\nu(y_i) = 0$.*

*Proof.* To prove (21), we choose a component index $i$ and show that the corresponding left-hand and right-hand side components equal each other for the two cases $|y_i| \geq \nu$ and $|y_i| < \nu$. First assume $|y_i| \geq \nu$. Then

$$\begin{aligned}
(s_\nu(y))_i &= \mathrm{sgn}(y_i)\max\{|y_i| - \nu, 0\} \\
&= \mathrm{sgn}(y_i)(|y_i| - \nu) \\
&= y_i - \nu\mathrm{sgn}(y_i) + \mathrm{sgn}(y_i)\cdot 0 \\
&= y_i - \nu\mathrm{sgn}(y_i) + \mathrm{sgn}(y_i)\,[\nu - |y_i|]^+ \\
&= (y - \nu\mathrm{sgn}(y) + \mathrm{sgn}(y)[\nu - |y|]^+)_i.
\end{aligned}$$

The second case, $|y_i| < \nu$, is immediately clear since both sides reduce to zero. □

**Lemma 3.2.** *The operator $s_\nu(\cdot)$ is component-wise non-expansive, i.e., for any $y^1, y^2 \in \mathbb{R}^n$,*

$$\left|s_\nu(y_i^1) - s_\nu(y_i^2)\right| \le \left|y_i^1 - y_i^2\right|, \; \forall i. \tag{22}$$

*Consequently, $s_\nu$ is non-expansive in any $\ell_p$ (quasi-)norms, for $p \ge 0$, and if $h$ is non-expansive in a given norm, then $s_\nu \circ h$ is as well. Moreover, consider the case when*

$$|s_\nu(y_i^1) - s_\nu(y_i^2)| = |y_i^1 - y_i^2|, \tag{23}$$

*which we refer to as the no-shrinkage condition. We have, for each index $i$:*

*i.* $(23) \implies \mathrm{sgn}(y_i^1) = \mathrm{sgn}(y_i^2),\ s_\nu(y_i^1) - s_\nu(y_i^2) = y_i^1 - y_i^2.$

*ii.* $(23)$ *and* $y_i^1 \ne y_i^2 \implies |y_i^1| \ge \nu,\ |y_i^2| \ge \nu$ *and* $|y_i^1| \ne |y_i^2|$.

*iii.* $(23)$ *and* $|y_i^2| < \nu \implies |y_i^1| < \nu,\ y_i^1 = y_i^2,\ s_\nu(y_i^1) = s_\nu(y_i^2) = 0.$

*iv.* $(23)$ *and* $|y_i^2| \ge \nu \implies |y_i^1| \ge \nu.$

*v.* $|y_i^2| \ge \nu$ *and* $\mathrm{sgn}(y_i^1) \ne \mathrm{sgn}(y_i^2) \implies |s_\nu(y_i^1) - s_\nu(y_i^2)| \le |y_i^1 - y_i^2| - \nu.$

*vi.* $s_\nu(y_i^1) \ne 0 = s_\nu(y_i^2) \implies |y_i^1| > \nu,\ |y_i^2| \le \nu,\ |s_\nu(y_i^1) - s_\nu(y_i^2)| \le |y_i^1 - y_i^2| - (\nu - |y_i^2|).$

The proof of this lemma, which is straightforward but rather lengthy and tedious, is given in the Appendix.

# 4 Convergence Analysis

In this section, we study the convergence of the fixed point iterations (20) as applied to our general $\ell_1$-regularized minimization problem (4) and the quadratic case (5). Assumption 1, which states that $f$ is a convex function with bounded Hessian in a neighborhood of an optimal solution of (4), is sufficient for our global convergence result and will apply throughout. Further assumptions (primarily on the rank of a particular minor of the Hessian of $f$) will be made to obtain linear convergence rate results in Section 4.2.

**Assumption 1.** *Problem* (4) *has an optimal solution set $X^* \ne \emptyset$, and there exists a set*

$$\Omega = \{x : \|x - x^*\| \le \rho\} \subset \mathbb{R}^n$$

*for some $x^* \in X^*$ and $\rho > 0$ such that $f \in C^2(\Omega)$, $H(x) := \nabla^2 f(x) \succeq 0$ for $x \in \Omega$ and*

$$\hat{\lambda}_{\max} := \max_{x \in \Omega} \lambda_{\max}(H(x)) < \infty. \tag{24}$$

For simplicity, we will use a constant parameter $\tau$ in the fixed point iterations (20): $x^{k+1} = s_\nu(x^k - \tau g(x^k))$, where $\nu = \tau/\mu$. In particular, we will always choose

$$\tau \in \left(0, 2/\hat{\lambda}_{\max}\right), \tag{25}$$

which guarantees that $h(\cdot) = I(\cdot) - \tau g(\cdot)$ is non-expansive in $\Omega$, and contractive in the range space of $H$ in the quadratic case. Our analysis can be extended to the case of variable $\tau$, but this would require more complicated notation and a reduction of clarity.

### 4.1 Global and Finite Convergence

From the mean-value theorem, we recall that for any $x, x' \in \Omega$

$$g(x) - g(x') = \left( \int_0^1 H(x' + t(x - x'))\, dt \right) (x - x') := \bar{H}(x, x')(x - x'). \tag{26}$$

In the lemma below, we verify that the operator $h$ is non-expansive and show that if applying $h$ to two points in $\Omega$ does not incur any contraction on their distance, then the gradients of $f$ are equal at the two points.

**Lemma 4.1.** *Under Assumption 1 and the choice of $\tau$ specified in* (25)*, $h(\cdot) = I(\cdot) - \tau g(\cdot)$ is non-expansive in $\Omega$, i.e., for any $x, x' \in \Omega$,*

$$\|h(x) - h(x')\| \leq \|x - x'\|. \tag{27}$$

*Moreover, $g(x) = g(x')$ whenever equality holds in* (27)*.*

*Proof.* Let $\bar{H} := \bar{H}(x, x')$. We first note that

$$h(x) - h(x') = x - x' - \tau(g(x) - g(x')) = (I - \tau \bar{H})(x - x').$$

Hence, in view of (25),

$$\begin{aligned} \|h(x) - h(x')\| &= \|(I - \tau \bar{H})(x - x')\| \\ &\leq \max\{|1 - \tau \lambda_{\max}(\bar{H})|, |1 - \tau \lambda_{\min}(\bar{H})|\} \|x - x'\| \\ &\leq \max\{|1 - \tau \hat{\lambda}_{\max}|, 1\} \|x - x'\| \\ &\leq \|x - x'\|. \end{aligned}$$

Let $s := x - x'$ and $p := \bar{H}^{1/2} s$. Then

$$\begin{aligned} \|h(x) - h(x')\| = \|x - x'\| &\iff \left\| s - \tau \bar{H} s \right\| = \|s\| \\ &\iff -2\tau\, s^T \bar{H} s + \tau^2 s^T \bar{H}^2 s = 0 \\ &\iff \tau p^T \bar{H} p = 2 p^T p \\ &\implies \tau \frac{p^T \bar{H} p}{p^T p} = 2 \text{ if } p \neq 0, \end{aligned}$$

which contradicts (39) since $\frac{p^T \bar{H} p}{p^T p} \leq \hat{\lambda}_{\max}$. Hence, $p = 0$, so that

$$g(x) - g(x') = \bar{H}^{1/2} p = 0,$$

which completes the proof. □

Since any two fixed points, say $x$ and $x'$, of the non-expansive mapping $s_\nu \circ h$ must satisfy the equality

$$\|x - x'\| = \|s_\nu \circ h(x) - s_\nu \circ h(x')\| = \|h(x) - h(x')\|, \tag{28}$$

Lemma 4.1 shows that $g(x) = g(x')$ as well. Hence, we have the following corollary and the first statement of Theorem 2.1.

**Corollary 4.1** (Constant optimal gradient). *Under Assumption 1, there is a vector $g^* \in \mathbb{R}^n$ such that*

$$g(x^*) \equiv g^*, \ \ \forall x^* \in X^* \cap \Omega. \tag{29}$$

We will make use of the following partition of all indices into $L$ and $E$, and obtain finite convergence for components in $L$ but linear convergence for components in $E$.

**Definition 1.** *Let $X^* \neq \emptyset$ be the solution set of* (4) *and $g^*$ be the vector specified in Corollary 4.1. Define*

$$L := \{i : \mu|g^*_i| < 1\} \ \ \textit{and} \ \ E := \{i : \mu|g^*_i| = 1\}. \tag{30}$$

It is clear from the optimality condition (10) that $L \cup E = \{1, 2, \cdots, n\}$,

$$\operatorname{supp}(x^*) \subseteq E, \text{ and } x^*_i = 0, \quad \forall i \in L, \ \forall x^* \in X^*. \tag{31}$$

There are examples in which $\operatorname{supp}(x^*) \subsetneq E$, so the two vectors $|x^*|$ and $\mathbf{1} - \mu|g^*|$ are always complementary but may not be strictly complementary.

The positive scalar $\omega$ defined below will also play a key role in the finite convergence property of the fixed-point iterations (which is proven in Theorem 4.1):

**Definition 2.** *Let $g^*$ be the vector specified in Corollary 4.1. Define*

$$\omega := \min\{\nu(1 - \mu|g^*_i|) : i \in L\} > 0. \tag{32}$$

In view of (31), we have for all $x^* \in X^*$ and all $i \in L$

$$\nu(1 - \mu|g^*_i|) = \nu - \tau|g^*_i| = \nu - |x^*_i - \tau g_i(x^*)| = \nu - |h_i(x^*)|,$$

and consequently, for any $x^* \in X^*$,

$$\omega = \min\{\nu - |h_i(x^*)| : i \in L\} > 0. \tag{33}$$

We now claim that Assumption 1 is sufficient for obtaining convergence of the fixed point iterations (20) and **finite convergence** for components in $L$.

**Theorem 4.1** (The general case). *Under Assumption 1, the sequence $\{x^k\}$ generated by the fixed point iterations* (20) *applied to problem* (4) *from any starting point $x^0 \in \Omega$ converges to some $x^* \in X^* \cap \Omega$. In addition, for all but finitely many iterations, we have*

$$x^k_i = x^*_i = 0, \qquad \forall i \in L, \tag{34}$$

$$\operatorname{sgn}(h_i(x^k)) = \operatorname{sgn}(h_i(x^*)) = -\mu g^*_i, \qquad \forall i \in E, \tag{35}$$

*where, for $\omega$ defined in* (32)*, the numbers of iterations not satisfying* (34) *and* (35) *do not exceed $\|x^0 - x^*\|^2/\omega^2$ and $\|x^0 - x^*\|^2/\nu^2$, respectively.*

The proof of Theorem 4.1 is rather lengthy and thus relegated to the next section.

In light of this theorem, every starting point $x^0 \in \Omega$ determines a converging sequence $\{x^k\}$ whose limit is a solution of (4). Generally, the solutions of (4) may be non-unique, as it is not difficult to construct simple examples for which different starting points lead to different solutions.

We recall that $x_E$ and $g^*_E$ are defined as the sub-vectors of $x$ and $g^*$ with components $x_i$ and $g^*_i$, $i \in E$, respectively. Without loss of generality, we assume $E = \{1, 2, \cdots, |E|\}$, and let $(x_E; \mathbf{0})$ denote the vector in $\mathbb{R}^n$ obtained from $x$ by setting the components $x_i$ for $i \in L$ equal to 0. The following corollary enables one to apply any convergence results for the gradient projection method to the fixed point iterations (20).

**Corollary 4.2.** *Under Assumption 1, after a finite number of iterations the fixed point iterations* (20) *reduce to gradient projection iterations for minimizing $\phi(x_E)$ over a constraint set $O_E$, where*

$$\phi(x_E) := -(g_E^*)^\top x_E + f((x_E; \mathbf{0})), \text{ and} \tag{36}$$

$$O_E = \{x_E \in \mathbb{R}^{|E|} : -\mathrm{sgn}(g_i^*)x_i \geq 0, \ \forall i \in E\}. \tag{37}$$

*Specifically, we have $x^{k+1} = (x_E^{k+1}; \mathbf{0})$ in which*

$$x_E^{k+1} := P_{O_E}\left(x_E^k - \tau \nabla\phi(x_E^k)\right), \tag{38}$$

*where $P_{O_E}$ is the orthogonal projection onto $O_E$, and $\nabla\phi(x_E) = -g_E^* + g_E((x_E, \mathbf{0}))$.*

*Proof.* From Theorem 4.1, there exists $K > 0$ such that for $k \geq K$ (34)-(35) hold. Let $k > K$. Since $x_i^k = 0$ for $i \in L$, it suffices to consider $i \in E$. For $i \in E$, we have $x_i^k \geq 0$ if $\mathrm{sgn}(h_i(x^{k-1})) = 1$ (equivalently, $g_i^* < 0$) and $x_i^k \leq 0$ if $\mathrm{sgn}(h_i(x^{k-1})) = -1$ ($g_i^* > 0$). Therefore, for any $i$ $(-g_i^* x_i^k) \geq 0$ for all $k > K$. Hence, $x^k \in O$ according to the definition (13) of $O$ and $x_E^k \in O_E$.

For $i \in E$, we calculate the quantity

$$\begin{aligned} y_i^{k+1} &:= x_i^k - \tau(\nabla\phi(x^k))_i \\ &= x_i^k - \tau(-g_i^* + g_i(x^k)) \\ &= h_i(x^k) - \nu\,(-\mu g_i^*) \\ &= (-\mu g_i^*)(|h_i(x^k)| - \nu) \\ &= \mathrm{sgn}(h_i(x^k))(|h_i(x^k)| - \nu), \end{aligned}$$

where (35) was used to obtain the last two expressions. Clearly, the fixed point iterations (20) restricted to the components $i \in E$ are

$$\left(x_E^{k+1}\right)_i = s_\nu \circ h_i(x^k) = \begin{cases} y_i^{k+1}, & -g_i^* y_i^{k+1} \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Equivalently,

$$\left(x_E^{k+1}\right)_i = \left(P_{O_E}(x_E^k - \tau\nabla\phi(x_E^k))\right)_i,$$

which completes the proof. □

Finally, a stronger global convergence result for convex quadratic functions, the cases in problems (2) and (5), follows directly from the general convergence result. We note that Assumption 1 is no longer necessary if the convex quadratic is bounded below. Due to the importance of the quadratic case, we state a separate theorem.

**Theorem 4.2** (The quadratic case)**.** *Let $f$ be a convex quadratic function that is bounded below, $H$ be its Hessian, and $\tau$ satisfy*

$$0 < \tau < 2/\lambda_{max}(H). \tag{39}$$

*Then the sequence $\{x^k\}$, generated by the fixed point iterations* (20) *from any starting point $x^0$, converges to some $x^* \in X^*$. In addition,* (34)–(35) *hold for all but finitely many iterations.*

## 4.2 Linear Rate of Convergence

We showed in Subsection 4.1 that $\{x^k\}$ generated by the fixed point iterations (20) converges to some point in $X^* \cap \Omega$. Throughout this subsection, we let

$$x^* := \lim_{k\to\infty} x^k,$$

and study the rate of convergence of $\{x^k\}$ to $x^*$ under different assumptions. Recall that a sequence $\{\|x^k - x^*\|\}$ converges to zero $q$-linearly if its $q_1$-factor is less than one, i.e., if

$$q_1 := \limsup_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} < 1;$$

while it is $r$-linearly convergent if it is bounded by a $q$-linearly convergent sequence.

As we will show, under appropriate assumptions $q$-linear convergence holds for any $\tau \in (0, 2/\hat{\lambda}_{\max})$. However, the $q_1$-factor may vary with different choices of $\tau$. In particular, we will consider choices of the form

$$\tau(\lambda) := \frac{\gamma(\lambda)}{\gamma(\lambda)+1}\frac{2}{\hat{\lambda}_{\max}}, \quad \gamma(\lambda) := \frac{\hat{\lambda}_{\max}}{\lambda}, \tag{40}$$

where $\hat{\lambda}_{\max}$ is defined in (24) and $\lambda > 0$ will take different values under different assumptions. It is easy to see that $\tau(\lambda) \in (0, 2/\hat{\lambda}_{\max})$ since $\gamma(\lambda) > 0$.

Some of our assumptions will involve the matrix $H_{EE}$:

**Definition 3.** *Let $H(x)$ denote the Hessian of $f$, and*

$$H_{EE}(x) := [H_{i,j}(x)]_{i,j\in E}$$

*denote the square sub-matrix of $H$ corresponding the index set $E$ defined in* (30).

We first make use of Corollary 4.2 by applying existing results for the gradient projection method to (38), which yields:

**Proposition 4.1.** *Let Assumption 1 hold. If (i) $H_{EE}(x^*)$ has full rank, or (ii) $f$ is defined as in* (8)*, then $\{\|x^k\|_1 + \mu f(x^k)\}$ converges to $\|x^*\| + \mu f(x^*)$ $q$-linearly and $\{x^k\}$ converges to $x^*$ $r$-linearly.*

Under the first condition, the above result follows from [41] and [33], while under the second condition it follows from [33]. However, by directly analyzing the original fixed point iterations, we can strengthen the convergence rate of $\{x^k\}$ from $r$-linear to $q$-linear. Theorem 4.3 does this under the assumption that $H_{EE}(x^*)$ is full rank; Theorem 4.4 assumes that $\operatorname{supp}(x^*) = E$ instead. We first define

$$\bar{H}^k \equiv \bar{H}(x^k, x^*) := \int_0^1 H(x^* + t(x^k - x^*))\, dt. \tag{41}$$

**Theorem 4.3.** *Let Assumption 1 hold, and assume that*

$$\lambda^E_{\min} := \lambda_{\min}(H_{EE}(x^*)) > 0. \tag{42}$$

*Then for any $\tau \in (0, 2/\hat{\lambda}_{\max})$, $\{x^k\}$ converges to $x^*$ $q$-linearly. Moreover, if $\tau$ is chosen as in* (40) *with $\lambda = \lambda^E_{\min}$, then the $q_1$-factor satisfies*

$$q_1 \le \frac{\gamma(\lambda^E_{\min}) - 1}{\gamma(\lambda^E_{\min}) + 1}. \tag{43}$$

*Proof.* Without loss of generality, we assume that all iteration counts, $k$, are large enough such that $x_i^k = x_i^* = 0$ for all $i \in L$, and that the spectrum of $\bar{H}_{EE}^k$ falls in the interval $[\lambda_{\min}^E - \epsilon, \hat{\lambda}_{\max}]$ for an arbitrary $\epsilon > 0$. (The first assumption on $k$ is valid because of the finite convergence properties of Theorem 4.1; the second follows from the continuity of the Hessian.) Since $x_i^k = x_i^* = 0$, $\forall i \in L$, the mean-value theorem yields

$$h_E(x^k) - h_E(x^*) = x_E^k - x_E^* - \tau(g_E(x^k) - g_E(x^*)) = (I - \tau\bar{H}_{EE}^k)(x_E^k - x_E^*).$$

Recall that $x^{k+1} = s_\nu \circ h(x^k)$ and $s_\nu(\cdot)$ is non-expansive. Hence,

$$\begin{aligned}\|x^{k+1} - x^*\| &\equiv \|x_E^{k+1} - x_E^*\| \\ &\leq \|h_E(x_E^k) - h_E(x_E^*)\| \\ &\leq \|I - \tau\bar{H}_{EE}^k\|\|x_E^k - x_E^*\| \\ &\leq \max\{|1 - \tau\hat{\lambda}_{\max}|, |1 - \tau\lambda_{\min}^E| + \tau\epsilon\}\|x_E^k - x_E^*\| \\ &\equiv \max\{|1 - \tau\hat{\lambda}_{\max}|, |1 - \tau\lambda_{\min}^E| + \tau\epsilon\}\|x^k - x^*\|.\end{aligned}$$

Clearly, $\max\{|1 - \tau\hat{\lambda}_{\max}|, |1 - \tau\lambda_{\min}^E| + \tau\epsilon\}$ is less than one for any $\tau \in (0, 2/\hat{\lambda}_{\max})$ and $\epsilon$ sufficiently small; In particular, it equals the right-hand side of (43) plus $\tau\epsilon$ with the choice $\tau = \tau(\lambda_{\min}^E)$. Since $\epsilon$ is arbitrary, (43) must hold. □

**Theorem 4.4.** *Let Assumption 1 hold, and also assume that $x^*$ satisfies (i) $\text{supp}(x^*) = E$ or, equivalently, that the strict complementarity condition*

$$|x^*| + (1 - \mu|g^*|) > 0, \tag{44}$$

*is satisfied, and (ii) the range space $\mathcal{R}(H_{EE}(x))$ of $H_{EE}(x)$ is invariant in a neighborhood $N^*$ of $x^*$. Whenever $H_{EE}(x^*) \neq \mathbf{0}$, let*

$$\lambda_{\min}^{\mathcal{R}} := \lambda_{\min}(V^\top H_{EE}(x^*)V) > 0, \tag{45}$$

*where $V$ is any orthonormal basis of $\mathcal{R}(H_{EE}(x^*))$.*

*If $H_{EE}(x^*) = \mathbf{0}$, then $x^k = x^*$ for all $k$ sufficiently large; otherwise $\{x^k\}$ converges to $x^*$ q-linearly for any $\tau \in (0, 2/\hat{\lambda}_{\max})$. In the latter case, if $\tau$ is chosen as in (40) with $\lambda = \lambda_{\min}^{\mathcal{R}}$, then the $q_1$-factor satisfies*

$$q_1 \leq \frac{\gamma(\lambda_{\min}^{\mathcal{R}}) - 1}{\gamma(\lambda_{\min}^{\mathcal{R}}) + 1}. \tag{46}$$

The proof of this theorem is given in Section 5.2. We note that $\mathcal{R}(H_{EE}(x))$ is invariant near $x^*$ if either $f$ is a quadratic function or $H_{EE}(x^*)$ has full rank.

Since Assumption 1 is not required in the proof of global convergence for convex quadratic $f$, we can directly derive the following results for this case, which is the situation one encounters with compressed sensing. The proof, which is similar to those of Theorems 4.3 and 4.4, is left to the reader.

**Corollary 4.3.** *Let $f$ be a convex quadratic function that is bounded below, and $\{x^k\}$ be the sequence generated by the fixed point iterations (20) with $\tau \in (0, 2/\lambda_{\max}(H))$.*

*i. If $H_{EE}$ has full rank, then $\{x^k\}$ converges to $x^*$ q-linearly. Moreover, if $\tau$ is chosen as in (40) with $\lambda = \lambda_{\min}(H_{EE})$, then the $q_1$-factor satisfies*

$$q_1 \leq \frac{\gamma(\lambda_{\min}(H_{EE})) - 1}{\gamma(\lambda_{\min}(H_{EE})) + 1}.$$

*ii. Let $x^*$ satisfy the strict complementarity condition (44). Then if $H_{EE} = \mathbf{0}$, $\{x^k\}$ converges to $x^*$ in a finite number of steps; otherwise $\{x^k\}$ converges to $x^*$ q-linearly. Moreover, if $\tau$ is chosen as in (40) with $\lambda := \lambda_{\min}(V^\top H_{EE} V)$, where $V$ is an orthonormal basis for the range space of $H_{EE}$, then the $q_1$-factor satisfies*

$$q_1 \leq \frac{\gamma(\lambda_{\min}(V^\top H_{EE} V)) - 1}{\gamma(\lambda_{\min}(V^\top H_{EE} V)) + 1}.$$

### 4.3 Discussion

The assumptions of Theorems 4.3 and 4.4 usually hold for compressed sensing reconstruction problems posed as in (5) or (2), in which case $A$ is often a Gaussian random matrix or has rows randomly chosen from an orthonormal matrix such as the FFT, DCT, or wavelets transform matrix. It is well-known that a randomly generated matrix is full rank with probability one (unless elements of the matrix are generated from a restricted space) [18]. Therefore, when $A \in \mathbb{R}^{m \times n}$ is a random matrix, the reduced Hessian for problem (5), i.e., $A_E^\top M A_E$, where $A_E$ consists of columns of $A$ with indices in $E$, will have a full rank with probability one as long as $|E| \leq m$, which is generally the case in practice. A similar argument can be made for partial orthonormal matrices. We believe that the strict complementarity assumption in Theorem 4.4 should also hold for random matrices with a prevailing probability, though we do not currently have a proof for this. We have observed the general convergence behavior predicted by our theorems empirically in computational studies.

In our convergence theorems, the choice of $\tau$ is restricted by the upper bound $2/\hat{\lambda}_{\max}$, where $\hat{\lambda}_{\max}$ is an upper bound for the largest eigenvalue of the Hessian. In compressed sensing applications, the quantity $\hat{\lambda}_{\max}$ is often easily obtained. When $A$ is a partial orthonormal matrix and $M = I$, then $\hat{\lambda}_{\max} = \lambda_{\max}(A^\top A) = 1$ and $\tau \in (0, 2)$. When $A$ is a standard Gaussian random matrix (with elements independently drawn from the standard normal distribution), then from well-known random matrix theory (see [25] or [18], for example) we have

$$n\left(1 - \sqrt{\frac{m}{n}}\right)^2 \leq \lambda_i(A^\top A) \leq n\left(1 + \sqrt{\frac{m}{n}}\right)^2$$

with prevailing probability for large $n$. In either case, upper bounding for $\tau$ is not an issue.

For simplicity, we have used a fixed $\tau \in (0, 2/\hat{\lambda}_{\max})$ in our analysis. However, this requirement could be relaxed in the later stage of the iterations when the actions of the mapping $h = I - \tau g$ concentrate on a "reduced space". In this stage, $h$ can remain contractive even if the upper bound on the Hessian is replaced by that on the reduced Hessian, which will generally increase the upper bound on $\tau$. For example, consider the quadratic problem (2) where $A$ is a partial orthonormal matrix. Then $\lambda_{\max}(A^\top A) = 1$, but $\lambda_{\max}(A_E^\top A_E) < 1$, such that $\tau$ can be chosen close to $2/\lambda_{\max}(A_E^\top A_E) > 2$, and $h$ remains contractive. Such a dynamic strategy, though theoretically feasible, is not straightforward to implement. It should be an interesting topic for further research.

## 5 Proofs of Convergence Results

In this section, Theorems 4.1 and 4.4 are proved through a sequences of technical results that lead to the final arguments.

### 5.1 Proof of Theorem 4.1

The following lemma establishes sufficient conditions for $x_i$, $i \in E$ and $x \in \Omega$, to be stationary with respect to the fixed point iterations (20).

**Lemma 5.1.** *Let Assumption 1 and the following four conditions hold for some $x \in \Omega$.*

*i.* $\|h(x) - h(x^*)\| = \|x - x^*\|$,

*ii.* $|h_i(x)| \geq \tau/\mu, \ \ \forall i \in E$,

*iii.* $\operatorname{sgn}(h_i(x)) = \operatorname{sgn}(h_i(x^*)), \ \ \forall i \in E$,

*iv.* $h_i(x) = h_i(x^*), \ \ \forall i \in L$.

*The first three conditions imply that*

$$x_i = (s_\nu \circ h(x))_i\,, \ \ \forall i \in E, \tag{47}$$

*and the last one implies that*

$$0 = (s_\nu \circ h(x))_i, \ \ \forall i \in L. \tag{48}$$

*Proof.* By Lemma 4.1, the first condition implies that $g(x) = g(x^*)$. We note that $\nu = \tau/\mu$. The proof of (47) is the following calculation, for which $i \in E$.

$$\begin{aligned} s_\nu \circ h_i(x) &= \operatorname{sgn}(h_i(x))(|h_i(x)| - \nu) \\ &= h_i(x) - \nu \operatorname{sgn}(h_i(x)) \\ &= x_i - \nu(\mu g_i^* + \operatorname{sgn}(h_i(x^*))) \\ &= x_i - \nu \cdot 0 \\ &= x_i, \end{aligned}$$

where the second equality follows from Condition 2, the fourth from Conditions 1 and 3, and the fifth from the optimality of $x^*$. In particular, the optimality of $x^*$ implies that

$$\operatorname{sgn}(h_i(x^*)) = \left\{ \begin{array}{cc} \operatorname{sgn}(x_i^*), & x_i^* \neq 0 \\ \operatorname{sgn}(0 - \tau g_i^*), & x_i^* = 0 \end{array} \right\} = \operatorname{sgn}(-g_i^*) \tag{49}$$

and, by the definition of $E$,

$$\mu g_i^* + \operatorname{sgn}(h_i(x^*)) = 0, \ \ \forall i \in E. \tag{50}$$

Finally, the last statement of the lemma is evident, since $|h_i(x^*)| = |\tau g_i(x^*)| < \nu$ for all $i \in L$. □

The four conditions of Lemma 5.1 do not involve the operator $s_\nu$, and are not sufficient to guarantee that $(s_\nu \circ h(x))_i = x_i$ for all $i \in L$. However, we show in the next lemma that a single non-contractiveness condition not only implies the four conditions in Lemma 5.1, but also ensures $(s_\nu \circ h(x))_i = x_i$ for all $i \in L$.

**Lemma 5.2.** *Under Assumption 1, if*

$$\|s_\nu \circ h(x) - s_\nu \circ h(x^*)\| \equiv \|s_\nu \circ h(x) - x^*\| = \|x - x^*\|, \tag{51}$$

*then $x$ is a fixed point (and therefore a solution of (4)), that is,*

$$x = s_\nu \circ h(x). \tag{52}$$

*Proof.* Recall that $s_\nu$ is component-wise non-expansive and $h$ is non-expansive in $\|\cdot\|$. Clearly, from (51),

$$\|x - x^*\| = \|s_\nu \circ h(x) - s_\nu \circ h(x^*)\| \leq \|h(x) - h(x^*)\| \leq \|x - x^*\|. \tag{53}$$

Hence, both inequalities hold with equality. In particular, the no-shrinkage condition (23) holds for $y^1 = h(x)$ and $y^2 = h(x^*)$, so the part i of Lemma 3.2 gives us

$$s_\nu \circ h(x) - s_\nu \circ h(x^*) = h(x) - h(x^*).$$

Rewriting this equation, we get

$$s_\nu \circ h(x) = x - \tau(g(x) - g(x^*)).$$

Finally, as the last inequality in (53) also holds with equality, we have $g(x) - g(x^*) = 0$ according to Lemma 4.1. □

The next lemma establishes the finite convergence properties of $\{x_i^k\}$ stated in Theorem 4.1.

**Lemma 5.3.** *Let Assumption 1 hold and let $\{x^k\}$ be generated by the fixed point iterations (20) starting from any $x^0 \in \Omega$. Then*

*i.* $x_i^k = 0\ \forall i \in L$ *for all but at most* $\|x^0 - x^*\|^2/\omega^2$ *iterations;*

*ii.* $\mathrm{sgn}(h_i(x^k)) = \mathrm{sgn}(h_i(x^*)) = -\mu g_i^*,\ \forall\ i \in E$, *for all but at most* $\|x^0 - x^*\|^2/\nu^2$ *iterations.*

*Proof.* We fix any $x^* \in X^*$ and consider $x_i^k \neq 0$ for some $i \in L$. In view of the non-expansiveness of $s_\nu(\cdot)$ and the related properties given in Lemma 3.2, we have

$$\begin{aligned} \left|x_i^{k+1} - x_i^*\right|^2 &= \left|s_\nu \circ h_i(x^k) - s_\nu \circ h_i(x^*)\right|^2 \\ &\leq \Big(|h_i(x^k) - h_i(x^*)| - (\nu - h_i(x^*))\Big)^2 \\ &\leq \left|h_i(x^k) - h_i(x^*)\right|^2 - \omega^2, \end{aligned}$$

where the last inequality follows from (33). The component-wise non-expansiveness of $s_\nu(\cdot)$ and the non-expansiveness of $h(\cdot)$ imply that

$$\|x^{k+1} - x^*\|^2 \leq \|h(x^k) - h(x^*)\|^2 - \omega^2 \leq \|x^k - x^*\|^2 - \omega^2.$$

Therefore, the number of iterations where $x_i^k \neq 0$ for some $i \in L$ cannot be more than $\|x^0 - x^*\|^2/\omega^2$. This proves the first statement.

For the second statement, we note that if $i \in \mathrm{supp}(x^*)$,

$$0 \neq x_i^* = \mathrm{sgn}(h_i(x^*)) \max\{|h_i(x^*)| - \nu, 0\},$$

so that $|h_i(x^*)| > \nu$ for $i \in \mathrm{supp}(x^*)$. On the other hand,

$$|h_i(x^*)| = \tau|g^*| = \tau/\mu = \nu,\ \forall i \in E \setminus \mathrm{supp}(x^*).$$

Therefore,

$$|h_i(x^*)| \geq \nu,\ \forall i \in E.$$

Now if $\mathrm{sgn}(h_i(x^k)) \neq \mathrm{sgn}(h_i(x^*))$ for some $i \in E$, then Lemma 3.2 implies

$$\begin{aligned}\left|x_i^{k+1} - x_i^*\right|^2 &= \left|s_\nu \circ h_i(x^k) - s_\nu \circ h_i(x^*)\right|^2 \\ &\leq \Big(|h_i(x^k) - h_i(x^*)| - \nu\Big)^2 \\ &\leq \left|h_i(x^k) - h_i(x^*)\right|^2 - \nu^2.\end{aligned}$$

Hence, the number of iterations for which $\mathrm{sgn}(h_i(x^k)) \neq \mathrm{sgn}(h_i(x^*))$ for some $i \in E$ cannot be more than $\|x^0 - x^*\|^2/\nu^2$. Moreover, it follows directly from (50) that $\mathrm{sgn}(h_i(x^*)) = -\mu g_i^*$ for all $i \in E$. □

**Proof of Theorem 4.1**

*Proof.* Based on these lemmas, we are now ready to prove Theorem 4.1. To show that $\{x^k\}$ converges, we (i) show that $\{x^k\}$ has a limit point, (ii) argue that it must be a fixed point because it satisfies the condition (51) of Lemma 5.2, and (iii) prove its uniqueness.

Since $s_\nu \circ h(\cdot)$ is non-expansive, $\{x^k\}$ lies in a compact subset of $\Omega$ and must have a limit point, say,

$$\bar{x} = \lim_{j\to\infty} x^{k_j}.$$

Since for any given fixed point $x^*$ the sequence $\{\|x^k - x^*\|\}$ is monotonically non-increasing, it has a limit which can be written as

$$\lim_{k\to\infty} \|x^k - x^*\| = \|\bar{x} - x^*\|, \tag{54}$$

where $\bar{x}$ can be any limit point of $\{x^k\}$. That is, all limit points, if more than one exists, must have an equal distance to any given fixed point $x^* \in X^*$.

By the continuity of $s_\nu \circ h(\cdot)$, the image of $\bar{x}$,

$$s_\nu \circ h(\bar{x}) = \lim_{j\to\infty} s_\nu \circ h(x^{k_j}) = \lim_{j\to\infty} x^{k_j+1},$$

is also a limit point of $\{x^k\}$. Therefore, from (54) we have

$$\|s_\nu \circ h(\bar{x}) - s_\nu \circ h(x^*)\| = \|\bar{x} - x^*\|,$$

which allows us to apply Lemma 5.2 to $\bar{x}$ and establish the optimality of $\bar{x}$.

By setting $x^* = \bar{x} \in X^*$ in (54), we establish the convergence of $\{x^k\}$ to its unique limit point $\bar{x}$:

$$\lim_{k\to\infty} \|x^k - \bar{x}\| = 0.$$

Finally, the finite convergence results (34)–(35) were proved in Lemma 5.3. □

## 5.2 Proof of Theorem 4.4

The next lemma gives a useful update formula for $k$ sufficiently large and $i \in \mathrm{supp}(x^*)$.

**Lemma 5.4.** *Under Assumption 1, after a finite number of iterations*

$$x_i^{k+1} = x_i^k - \tau(g_i(x^k) - g_i^*), \quad \forall i \in \mathrm{supp}(x^*). \tag{55}$$

*Proof.* Since $x^k \to x^* \in X^* \cap \Omega$ and $h(\cdot)$ is continuous component-wise, $h_i(x^k) \to h_i(x^*)$. The fact that $|h_i(x^*)| > \nu$ for $i \in \text{supp}(x^*)$ implies that after a finite number of iterations we have $|h_i(x^k)| > \nu$ for $i \in \text{supp}(x^*)$. This gives

$$\begin{aligned} x_i^{k+1} &= \text{sgn}(h_i(x^k))(|h_i(x^k)| - \nu) \\ &= h_i(x^k) - \nu\,\text{sgn}(h_i(x^k)) \\ &= x_i^k - \tau g_i(x^k) - (\tau/\mu)\,(-\mu g_i^*) \\ &= x_i^k - \tau(g_i(x^k) - g_i^*), \end{aligned}$$

for any $i \in \text{supp}(x^*)$. □

**Proof of Theorem 4.4**

*Proof.* Without loss of generality, we can assume that $k$ is sufficiently large so that (55) holds and $x^k \in N^*$, where $N^*$ is defined in Theorem 4.4. Since $x_i^k = 0$ for any $i \in L$, it suffices to consider the rate of convergence of $x_i^k$ for $i \in E = \text{supp}(x^*)$, where equality follows from the strict complementarity assumption on $x^*$.

Let $\bar{H}^k$ be defined as in (41). By assumption, the range and null spaces of $\bar{H}_{EE}^k$ are now invariant for all $k$. Let $P = VV^\top \in \mathbb{R}^{|E|\times|E|}$, the orthogonal projection onto the range space of $H_{EE}(x^*)$. Then $I - P$ is the orthogonal projection onto the null space of $H_{EE}(x^*)$. Also recall that $x_E$ denotes the sub-vector of $x$ corresponding to the index set $E$.

Since $E = \text{supp}(x^*)$, Lemma 5.4 implies that

$$x_E^{k+1} = x_E^k - \tau(g(x^k) - g(x^*))_E = x_E^k - \tau\bar{H}_{EE}^k(x_E^k - x_E^*). \tag{56}$$

At each iteration the update, $-\tau\bar{H}_{EE}^k(x_E^k - x_E^*)$, stays in the range space of $H_{EE}(x^*)$. This implies that the null space components of the iterates have converged to the null space components of $x^*$, namely, for all $k$ sufficiently large,

$$(I - P)(x_E^k - x_E^*) \equiv \mathbf{0}. \tag{57}$$

If $H_{EE}(x^*) = 0$, then the range space is empty and the update vanishes such that $x^k = x^*$ after a finite number of steps.

Now we assume that $H_{EE}(x^*) \neq 0$ so that $\lambda_{\min}^{\mathcal{R}} > 0$ exists. It suffices to consider the rate of convergence of $\{Px_E^k\}$ to $Px_E^*$. It follows from (56) and (57) that

$$x_E^{k+1} - x_E^* = P(x_E^{k+1} - x_E^*) = P(I - \tau\bar{H}_{EE}^k)P(x_E^k - x_E^*). \tag{58}$$

By a routine continuity argument, we know that there exists an arbitrarily small constant $\epsilon > 0$ such that for all $k$ sufficiently large the eigenvalues of $V^\top H_{EE}^k V$ satisfy

$$\hat{\lambda}_{\max} \geq \lambda_i\left(V^\top H_{EE}^k V\right) \geq \lambda_{\min}^{\mathcal{R}} - \epsilon > 0, \ \ \forall i.$$

Consequently, given the definition of $\tau$ in (40) and noting that $P^2 = P = VV^\top$, we calculate from (58):

$$\begin{aligned} \|x_E^{k+1} - x_E^*\| &\leq \|P(I - \tau\bar{H}_{EE}^k)P\|\,\|x_E^k - x_E^*\| \\ &= \|I - \tau V^\top\bar{H}_{EE}^k V\|\,\|x_E^k - x_E^*\| \\ &= \max\left\{|1 - \tau\hat{\lambda}_{\max}|, |1 - \tau\lambda_{\min}^{\mathcal{R}}| + \tau\epsilon\right\}\|x_E^k - x_E^*\| \\ &= \left(\frac{\gamma(\lambda_{\min}^{\mathcal{R}}) - 1}{\gamma(\lambda_{\min}^{\mathcal{R}}) + 1} + \tau\epsilon\right)\|x_E^k - x_E^*\|, \end{aligned} \tag{59}$$

which implies (46) since $\epsilon$ can be arbitrarily small. □

# 6 A Continuation Method

Our algorithm for solving (4), that is,

$$\min_{x\in\mathbb{R}^n} \|x\|_1 + \mu f(x), \tag{60}$$

consists of applying the fixed-point iterations

$$x^{k+1} = s_\nu \circ h(x^k) := \mathrm{sgn}(x^k - \tau g(x^k)) \odot \max\{|x^k - \tau g(x^k)| - \nu, 0\},\ \mu\nu = \tau$$

(see (20) and (25)) within the continuation (or path-following) framework described below. Further extensions that may improve our algorithm are certainly possible, but are beyond the the scope of this paper. Indeed, a strength of this algorithm is that a simple implementation is sufficient to obtain good results. Our implementation will be fully described in the next section.

## 6.1 Homotopy Algorithms in Statistics

Statisticians often solve (2) (which is (4) with $f(x) = \frac{1}{2}\|Ax - b\|^2$) in the context of regression. In Bayesian terminology, this corresponds to maximizing the *a posteriori* probability for recovering the signal $x$ from the measurement $b = Ax + \epsilon$, where the prior on $x$ is Laplacian and $\epsilon$ is Gaussian white noise. Practically, such a procedure may be preferred over standard least squares because a sparse solution of (2) explicitly identifies the most significant regressor variables.

As intimated in the Introduction, variations on (2) may be used in different applications and contexts. For example, problem (2) is closely related to this quadratically constrained $\ell_1$-minimization problem:

$$\min_x \left\{ \|x\|_1 \;\middle|\; \frac{1}{2}\|Ax - b\|^2 \le \sigma^2 \chi^2_{1-\alpha,m} \right\}, \tag{61}$$

which is often used when an estimated noise level $\sigma$ is available. Alternatively, one can constrain $\|x\|_1$ and minimize the sum of squares of the residual $Ax - b$:

$$\min_x \left\{ \frac{1}{2}\|Ax - b\|^2 \;\middle|\; \|x\|_1 \le t \right\}. \tag{62}$$

Statisticians often refer to the above problem as the Least Absolute Shrinkage and Selection Operator (LASSO) [49].

Problems (2), (61) and (62) are equivalent in the sense that once the value of one of $\mu$, $\sigma$, or $t$ is fixed, there are values for the other two quantities such that all three problems have the same solution. For a detailed explanation, please see [44].

Least Angle Regression (LARS) (see [19], for example) is a method for solving (62). LARS starts with the zero vector and gradually increases the number of nonzeros in the approximation $x$. In fact it generates the full path of solutions that results from setting the right-hand side of the constraint to every value in the interval $[0, t]$. Thus, LARS is a homotopy algorithm. The construction of the path of solutions is facilitated by the fact that it is piecewise linear, such that any segment can be generated given the solutions at turning points, which are the points at which at least one component changes from zero to non-zero or vice versa. Thus LARS and other homotopy algorithms [35, 40, 55] solve (62) by computing the solutions at the turning points encountered as $t$ increases from 0 to a given value. These algorithms require the solution of a least squares problem at every iteration, where the derivative matrix of the residuals consists of the columns of $A$ associated with the nonzero components of the current iterate. For large scale problems, solving these intermediate least squares problems may prove costly, especially when $A$ is a partial fast transform matrix that is not stored explicitly, and/or the solution is only moderately sparse.

We found it helpful for our algorithm to adopt a continuation strategy similar to homotopy in the sense that an increasing sequence of $\mu$ values is used. However, our algorithm does not track turning points or solve any least squares sub-problems, and so only approximately follows the solution path.

### 6.2 A Continuation Strategy

The convergence analysis indicates that the speed of the fixed point algorithm is determined by the values of $\nu = \tau/\mu$ and $\omega$ (see Theorem 4.1), and the spectral properties of the Hessian of $f(x)$ (see Theorems 4.3 and 4.4). The signs of $h_i(x^k)$ evolve to agree with those of $h_i(x^*)$ for $i \in E$ faster for larger $\nu$ (equivalently, for smaller $\mu$). Similarly, large $\omega$ implies fast convergence of the $|x_i^k|$, $i \in L$, to zero. Once the finite convergence properties of Theorem 4.1 are satisfied, all action is directed towards reducing the errors in the $E$ components, and the (worst-case) convergence rate is dictated by $\|I - \tau \bar{H}_{EE}\|$, which can be considerably smaller than $\|I - \tau \bar{H}\|$, especially when $|E| \ll n$.

In general, we have little or no control over the value of $\omega$, nor the spectral properties of the Hessian. On the other hand, we do have the freedom to choose $\tau$ and $\nu = \tau/\mu$. For fixed $\tau$ we found $\tau \in [1/\hat{\lambda}_{\max}, 2/\hat{\lambda}_{\max})$ to be superior to $\tau \in (0, 1/\hat{\lambda}_{\max})$. Beyond this, $\tau$ does not have much effect on $\nu$ and can be chosen empirically or based on considerations concerning $||I - \tau \bar{H}_{EE}||$. $\mu$, on the other hand, while it must eventually be equal to some specified value $\bar{\mu}$, can in the meantime be chosen freely to produce a wide range of $\nu$ values. Thus, since larger $\nu$ are desired, we propose a continuation strategy for $\mu$. In particular, if problem (60) is to be solved with $\bar{\mu}$, we propose solving a sequence of problems (60) defined by an increasing sequence $\{\mu_j\}$, as opposed to fixing $\nu = \tau/\bar{\mu}$. When a new problem, associated with $\mu_{j+1}$, is to be solved, the approximate solution for the current ($\mu_j$) problem is used as the starting point. In essence, this framework approximately follows the path $x^*(\mu)$ in the interval $[\mu_1, \bar{\mu}]$, where for any given $\mu$ value $x^*(\mu)$ is an optimal solution for (60). This path is well-defined if the solution to (60) is unique for $\mu \in [\mu_1, \bar{\mu}]$. Even if this is not the case, it is reassuring to observe that the algorithm itself is well-defined. A formal statement of our fixed point continuation method is given in Algorithm 1.

Our computational experience indicates that the performance of this continuation strategy is generally superior to that of directly applying the fixed point iterations (20) with the specified value $\bar{\mu}$. This is also in line with the observations of [14, 49, 40, 35]. Moreover, since $x^*(\mu)$ tends to be sparser for smaller $\mu$, the reduced Hessian $\bar{H}_{EE}$ tends to be smaller and better conditioned in this case, such that the continuation strategy should improve the convergence rate for those components in $E$ in addition to the rate of finite convergence of $\operatorname{sgn}(h_i(x^k))$, $i \in E$.

In principle, our fixed point continuation algorithm can be used to solve problems (61) and

**Algorithm 1** Fixed Point Continuation (FPC) Algorithm

```
Require: $A$, $b$, $x^0$, and $\bar{\mu}$
Select $0 < \mu_1 < \mu_2 < \cdots < \mu_L = \bar{\mu}$. Set $x = x^0$.
for $\mu = \mu_1, \mu_2, \cdots, \mu_L$ do
    while "not converged" do
        Select $\tau$ and set $\nu = \tau/\mu$
        $x \leftarrow s_\nu \circ h(x)$
    end while
end for
```

(62) in addition to (60). Take the LASSO problem (62) as an example. When we start our algorithm with a small $\mu$ value, the corresponding optimal $\|x\|_1$ value is also small. As we gradually increase $\mu$, the optimal $\|x\|_1$ value increases. We can stop the process once $\|x\|_1$ approximately equals $t$, backtracking if necessary. As interesting as such extensions may be, they are not in the scope of the current paper.

# 7 Numerical Results

We demonstrate the effectiveness of Algorithm 1 for the case of compressed sensing, that is for solving

$$\min_{x \in \mathbb{R}^n} \|x\|_1 + \frac{\bar{\mu}}{2}\|Ax - b\|_M^2, \tag{63}$$

where the second term enforces approximate satisfaction of $Ax = b$, an underdetermined linear system, and $b$ was generated from a sparse signal $x_s$. In particular, $A \in \mathbb{R}^{m \times n}$, $m \leq n$, $k = |\text{supp}(x_s)| \leq m$, and

$$b = A(x_s + \epsilon_1) + \epsilon_2, \tag{64}$$

where $\epsilon_1$ ($\epsilon_2$) is a vector whose elements are *i.i.d.* distributed as $N(0, \sigma_1^2)$ ($N(0, \sigma_2^2)$), that is, both the signal and the measurements may be corrupted with Gaussian noise. As described in Section 1.1, $M$ is an $m \times m$ weighting matrix. If $M = I$ then the second term of (63) becomes the familiar least squares objective $\frac{\mu}{2}\|Ax - b\|_2^2$. The above compressed sensing model (64) suggests a particular form for $M$ from which we derive appropriate values of $\mu$. We also present relevant simplifications and scalings.

Our results are restricted to two types of $A$ matrices: Gaussian matrices whose elements are *i.i.d.* standard normal (`randn(m,n)` in Matlab), and partial DCT matrices whose $m$ rows are chosen at random (uniformly) from the $n \times n$ discrete cosine transform (DCT) matrix; both are "good" matrices for compressed sensing. While Gaussian matrices are explicitly stored, partial DCT matrices are implicitly stored fast transforms for which matrix-vector products cost just $O(n \log n)$ flops. Furthermore, partial DCT matrices have orthonormal rows such that $AA^T = I$.

## 7.1 Selection of $M$ and $\mu$

The statistical model (64) implies that the error $Ax_s - b$ is normally distributed with zero mean and covariance $\sigma_1^2 AA^T + \sigma_2^2 I$ such that

$$\text{Prob}\left((Ax_s - b)^T(\sigma_1^2 AA^T + \sigma_2^2 I)^{-1}(Ax_s - b) \leq \chi^2_{1-\alpha,m}\right) = 1 - \alpha, \tag{65}$$

where $\chi^2_{1-\alpha,m}$ is the $1-\alpha$ critical value of the $\chi^2$ distribution with $m$ degrees of freedom. Therefore, it is appropriate to try to recover $x_s$ by solving (5) with

$$M = (\sigma_1^2 AA^T + \sigma_2^2 I)^{-1}, \tag{66}$$

which is a positive definite matrix as long as $\sigma_2$ is non-zero or $\sigma_1$ is non-zero and $AA^T$ is nonsingular.

An estimate for $\mu$ may be derived by considering the optimality conditions for (63), see (10), which imply that for any solution $x$ of (63), $\|g(x)\|_\infty = \left\|A^T M(Ax-b)\right\|_\infty \le 1/\mu$. Then since $\|\cdot\|_2 \le \sqrt{n}\,\|\cdot\|_\infty$ and $\left\|A^T M(Ax-b)\right\|_2^2 = \|Ax-b\|^2_{MAA^TM}$, if we let

$$\underline{\sigma}^2 = \lambda_{\min}(M^{1/2}AA^TM^{1/2}),$$

we have

$$\underline{\sigma}^2 \|Ax-b\|_M^2 \le \|Ax-b\|^2_{MAA^TM} \le \frac{n}{\mu^2}, \tag{67}$$

such that setting

$$\mu = \frac{1}{\underline{\sigma}}\sqrt{\frac{n}{\chi^2_{1-\alpha,m}}} \tag{68}$$

guarantees that $\|Ax-b\|_M^2 \le \chi^2_{1-\alpha,m}$ at optimality with probability $1-\alpha$ or greater. Note that this $\mu$ is well-defined as long as $A$ and $M$ are full rank, since this gives $\underline{\sigma}^2 > 0$.

In two important cases it is possible to set $M = I$. For simplicity of implementation we also scale the quadratic term so that the condition $0 < \tau < 2/\lambda_{\max}(A^TMA)$, (39), becomes $0 < \tau < 2$. This yields three cases.

Case 0. When $\sigma_1 > 0$ and $AA^T \ne I$, $M$ must be explicitly constructed, and we require estimates of $\overline{\sigma}^2 = \lambda_{\max}(M^{1/2}AA^TM^{1/2})$ and $\underline{\sigma}^2 = \lambda_{\min}(M^{1/2}AA^TM^{1/2})$ (obtainable with Lancozs iterations). Then an equivalent problem to (63) with $M$ and $\mu$ defined in (66) and (68) is obtained by substituting the following quantities for $\mu$, $A$ and $b$:

$$\mu = \frac{\overline{\sigma}^2}{\underline{\sigma}}\sqrt{\frac{n}{\chi^2_{1-\alpha,m}}},\ \tilde{A} = \frac{1}{\overline{\sigma}}A,\ \tilde{b} = \frac{1}{\overline{\sigma}}b. \tag{69}$$

This yields $\lambda_{\max}(\tilde{A}^TM\tilde{A}) = 1$, as desired, since the nonzero eigenvalues of $M^{1/2}AA^TM^{1/2}$ and $A^TMA$ are equal.

Case 1. If $\sigma_1 = 0$ and $AA^T \ne I$, we set $M = I$, $\overline{\sigma}^2 = \lambda_{\max}(AA^T)$, $\underline{\sigma}^2 = \lambda_{\min}(AA^T)$ and

$$\mu = \frac{\overline{\sigma}^2}{\sigma_2\underline{\sigma}}\sqrt{\frac{n}{\chi^2_{1-\alpha,m}}},\ \tilde{A} = \frac{1}{\overline{\sigma}}A,\ \tilde{b} = \frac{1}{\overline{\sigma}}b. \tag{70}$$

Note that for Gaussian $A$ matrices we have the bounds

$$\left(1-\sqrt{\frac{m}{n}}\right)^2 n \le \underline{\sigma}^2 \le \overline{\sigma}^2 \le \left(1+\sqrt{\frac{m}{n}}\right)^2 n,$$

with a prevailing probability, which become tight as $n \to \infty$.

Case 2. If $AA^T = I$, then by setting $M = I$ and

$$\mu = \frac{1}{\sqrt{\sigma_1^2+\sigma_2^2}}\sqrt{\frac{n}{\chi^2_{1-\alpha,m}}}, \tag{71}$$

we automatically arrive at $0 < \tau < 2$, without any scaling of $A$ and $b$.

## 7.2 Implementation

In addition to $M$ and $\mu$, FPC (Algorithm 1) requires an initial guess $x^0$; a method for choosing $\mu_1$, $\mu_2$, $\ldots$, $\mu_{L-1}$; a convergence criterion; and a method for choosing $\tau$. Below we describe our default choices for these quantities, and discuss a post-processing procedure.

### 7.2.1 Basic Algorithm

Since the performance of compressed sensing algorithms varies with the number of measurements, $m$, and the number of non-zeros of $x_s$, $k$, algorithm parameters are best evaluated by computing phase plots like in, that is, plots that depict some performance statistic as a function of

$$\delta = m/n \text{ and } \rho = k/m, \tag{72}$$

as is done in [15]. In this work, we recorded mean statistics on the speed and accuracy of FPC at a number of equally spaced grid points $(\delta, \rho) \in [0,1] \times [0,1]$ for each parameter combination of interest, and used the results to parameterize FPC and set default values. The results are described below, but it may be possible to improve upon the cited values as our implementation continued to evolve after the defaults were set.

Experiments showed that if $\tau$ is fixed, $\tau$ should be large (near 2) 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 we set $\tau$ according to

$$\tau = \min\{1 + 1.665(1-\delta), 1.999\}. \tag{73}$$

We set the initial iterate to $x^0 = \tau A^T M b$ since this quantity contains problem-specific information and is simple to compute. Also note that it is the minimum norm solution to $\min \|Ax - b\|^2$ when $AA^T = I = M$ and $\tau = 1$.

We initialize $\mu_1$ so that $\frac{\tau}{\mu_1} = \gamma \|x^0\|_\infty$, where $0 < \gamma < 1$ is a user-defined constant. This choice ensures that $s_\nu$ will shrink most components of $h(x)$ to 0; only components larger than $\gamma\|x^0\|_\infty$ will remain non-zero. We also found it convenient and effective to set $\mu_2, \ldots, \mu_L$ according to the modified geometric progression $\mu_i = \min\{\mu_1 \beta^{i-1}, \bar{\mu}\}$, where $\beta > 1$ is a suitable multiplier and $L$ is the first integer $i$ for which $\mu_i = \bar{\mu}$. Our default choices for $\gamma$ and $\beta$ are $\gamma = 0.99$ and $\beta = 4$. In general, $\gamma$ did not impact the accuracy of FPC, but $\gamma = 0.99$ resulted in faster CPU times as compared to $\gamma = 0.5$ and $\gamma = 0.9$. We also found $\gamma = 0.9$ and $\beta = 2$ to be an effective pairing, which suggests that $\gamma$ and $\beta$ should be adjusted so as to keep $L$ approximately constant.

The outer iteration corresponding to $\mu_i$ is terminated when

$$\frac{\|x^{k+1} - x^k\|}{\max\{\|x^k\|, 1\}} < xtol \quad \text{and} \quad \mu_i \|g(x^k)\|_\infty - 1 < gtol.$$

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

We are now in a position to comment on the computational complexity of Algorithm 1 (FPC) as applied to compressed sensing. The approximate amount of storage (number of array elements) and number of floating point operations (flops) per inner iteration are listed in

Table 7.2.1 for three situations of interest. In the first two, the $mn$ elements of $A$ are explicitly stored and matrix-vector multiplications cost $O(mn)$ flops. The third case refers to partial fast transform matrices for which $A$ is identified by listing the $m$ rows of the $n \times n$ transform matrix (FFT, DCT, etc.) that were used to compute $b$. In this case, matrix-vector operations cost $O(n \log n)$, which is the cost of a full transform.

Table 1: Complexity of Algorithm 1 for three types of compressed sensing problems.

| Problem Type | Required Storage | Flops per Iteration |
|---|---|---|
| Explicit $A$, $M = I$ | $mn + O(n)$ | $O(mn)$ |
| Explicit $A$, $M \neq I$ | $mn + m^2 + O(n)$ | $O(mn)$ |
| Fast transform $A$, $M = I$ | $O(n)$ | $O(n \log n)$ |

Of course the total cost of Algorithm 1 depends on the number of inner iterations, a quantity that varies with $\delta$, $\rho$, $\bar{\mu}$, and problem instance. For compressed sensing we found that convergence usually occurs within 1000 iterations, most times far less, when the original signal is sparse enough to be accurately reconstructed. Furthermore, the number of iterations is approximately independent of the signal length, $n$.

### 7.2.2 De-biasing

Unless otherwise stated, all of the FPC results are the output of Algorithm 1 applied to (63) with the recommended values of $M$ and $\mu$. However, since the original signal is truly sparse in this work, the accuracy of the reconstruction was often improved by post-processing the results of Algorithm 1 with the de-biasing procedure of Algorithm 2. This procedure is similar to the post-processing step recommended by the designers of GPSR [23], and to the calculation of $x^k$ during each iteration of StOMP [15].

Based on the $\mu$ and timing studies below, and an implementation of Algorithm 2 that used $tol = 3\sigma$, where $\sigma =$ `sqrt(mean(diag(M)))` in the most general case, it seems to be advantageous to use the de-biasing procedure with all DCT-$A$ problems, and with Gaussian-$A$ problems for which there is no signal noise ($\sigma_1 = 0$). Furthermore, when there is very little noise in the system we recommend using the de-biasing algorithm in combination with a smaller value of $\bar{\mu}$, as this recovers most of the accuracy at a reduced cost. In the following it is sometimes appropriate to refrain from de-biasing in order to obtain more meaningful comparisons. Therefore we will state which results reflect de-biasing and which do not.

**Algorithm 2** De-Biasing

```
Require: $A$, $b$, and $x$ from Algorithm 1
$S = \{i \mid |x_i| > tol\}$
if $1 \leq |S| \leq m$ then
    $Z = \{1, \ldots, n\} \setminus S$
    $x_Z = 0$
    $x_S = \arg\min_x \|A_S x - b\|_2^2$
end if
```

### 7.3 Computational Details

In addition to the two types of $A$ matrices, Gaussian and DCT, we considered four noise scenarios. The resulting eight combinations are listed in Table 7.3 along with their classifications (in the last column) according to Section 7.1 and abbreviations that will be used throughout the remainder of this section. The original signal for a given problem instance was generated by choosing the locations of $x_s$'s $k$ nonzeros uniformly at random, and then taking those nonzero values from the $N(0, 4)$ distribution. $b$ was calculated according to (64).

Table 2: The eight compressed sensing scenarios studied in this section.

| Abbrev. | Matrix Type | Noise Scenario (N*) | $\sigma_1$ | $\sigma_2$ | Case |
|---|---|---|---|---|---|
| GN1 | Gaussian (G*) | 1 - small measurement | 0 | 1E-8 | 1 |
| GN2 | | 2 - medium on both | 5E-3 | 1E-3 | 0 |
| GN3 | | 3 - medium measurement | 0 | 5E-3 | 1 |
| GN4 | | 4 - large signal | 1E-2 | 1E-8 | 0 |
| DN1 | DCT (D*) | 1 - small measurement | 0 | 1E-8 | 2 |
| DN2 | | 2 - medium on both | 5E-3 | 1E-3 | 2 |
| DN3 | | 3 - medium measurement | 0 | 5E-3 | 2 |
| DN4 | | 4 - large signal | 1E-2 | 1E-8 | 2 |

All calculations were completed using Matlab 7.3 on a Dell Optiplex GX620 with a 3.2 GHz processor and 4 GB RAM. Normally and uniformly distributed pseudo-random numbers were generated as needed using the Matlab commands `randn` and `rand`.

Below we compare the performance of Algorithm 1 (Fixed Point Continuation or FPC) to three other recently proposed compressed sensing algorithms: StOMP, GPSR-BB and l1_ls. As described in Section 1.3, StOMP does not solve (63) at all, but uses a greedy approach to reconstruct $x_s$. GPSR-BB and l1_ls were designed for problems equivalent to (63), but only if $M = I$. This inconsistency was corrected by multiplying $A$ and $b$ on the left by $M^{1/2}$ as appropriate.

StOMP, on the other hand, relies on the scaling of the $A$ matrix, and so, while there is no discrepancy with regards to the DCT scenarios, the G* scenarios for StOMP do not correspond exactly to those shown for the other three algorithms. In particular, in order for its thresholding schemes to work properly, StOMP requires Gaussian $A$ matrices to be further processed so that either (a) the columns are scaled to unit norm or (b) a matrix with orthonormal rows is obtained via QR factorization [15]. We chose option (a) for this work, that is, G* for StOMP refers to solving (63) with $M = I$ and a Gaussian $A$ matrix whose columns have been scaled to unit norm.

All four algorithms use fast transforms instead of explicit matrix operations when $A$ is a partial DCT matrix. In particular, for StOMP we used its iterative solver option to solve the required least squares problems.

### 7.4 Quality of Compressed Sensing Reconstructions

Similar to [15], we evaluated the suitability of Algorithm 1 for compressed sensing by generating phase plots for each of the cases listed in Table 7.3. A selection is presented in Figure 1; similar phase plots for StOMP are in Figure 2. Both sets of phase plots depict the number of

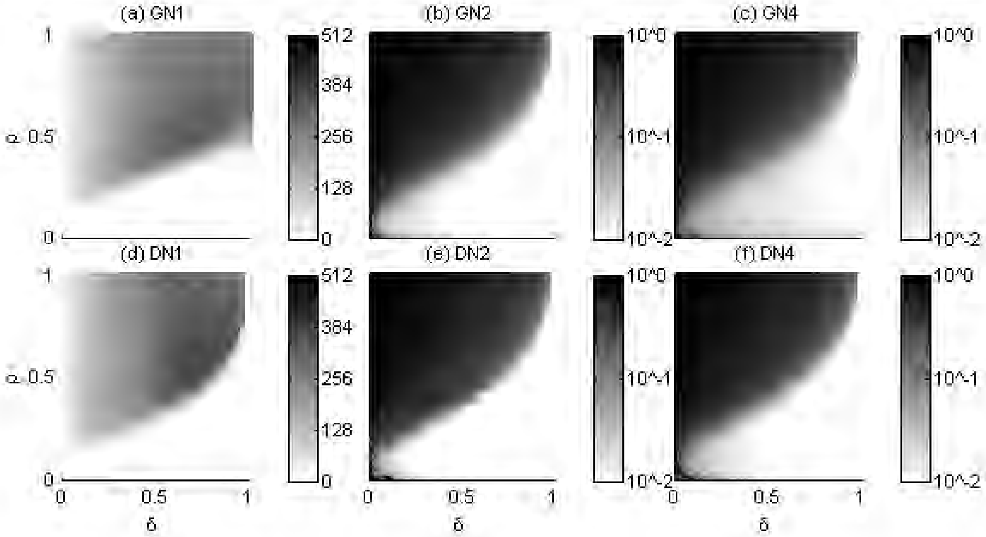

Figure 1: Phase plots for Algorithm 1 (FPC). Subplots (a) and (d) depict the number of elements in the original signal that were not recovered within 1E-4 relative error. The intensities in subplots (b-c) and (e-f) indicate the relative 2-norm error between the recovered signal and the original signal. Black is used for any relative 2-norm error greater than or equal to 1; quantities less than or equal to 1E-2 are shaded white. In all cases $\delta = m/n$, the ratio of measurements to signal length and $\rho = k/m$, the number of nonzeros in the original signal divided by the number of measurements.

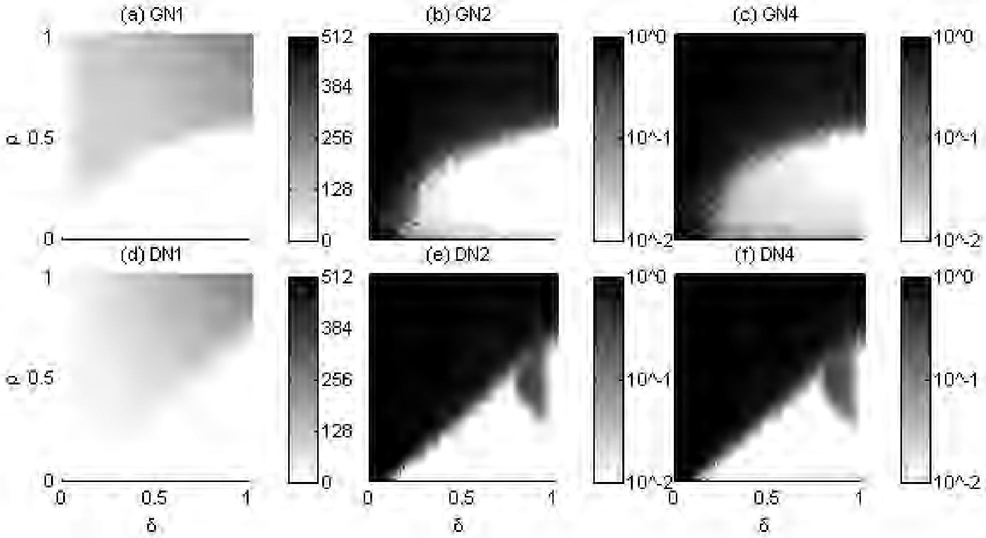

Figure 2: Phase plots for StOMP. This figure is analogous to Figure 1.

components not recovered to a relative tolerance of 1E-4 in the *N1 cases (with little noise) and the relative 2-norm error $\|x - x_s\|/\|x_s\|$ in the remaining cases (with substantial noise). Each $(\delta, \rho)$ point in each phase plot represents an average over 30 runs with $n = 512$.

All of the phase plots show that the original signal is recovered well as long as it is sparse enough and there are enough measurements. However, the location of the transition curve,

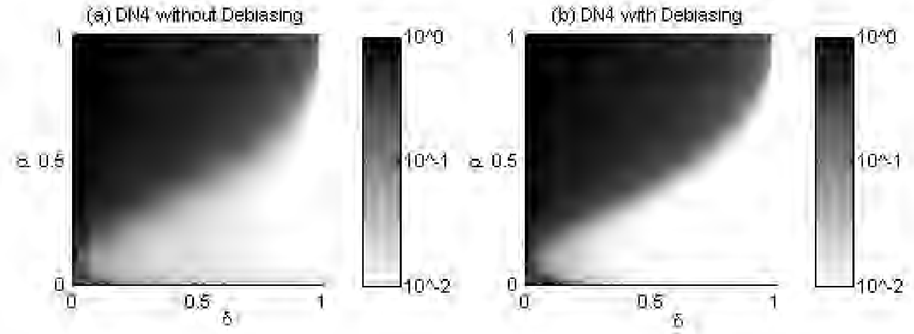

Figure 3: Phase plots for FPC, Scenario DN4 ($\sigma_1$ = 1E-2, $\sigma_2$ = 1E-8). Plot (a) shows the relative 2-norm error for the FPC results without debiasing. (b) is analogous, but includes de-biasing, that is, post-processing with Algorithm 2.

and its qualitative shape, depends on the recovery algorithm, type of $A$ matrix and noise level.

The FPC results for GN1 and DN1 are the result of setting $\bar{\mu} = 5000$ and applying the de-biasing procedure to the FPC output. $\mu$ was set to its recommended value for the remainder of the plots, however, the DN2 and DN4 results also include de-biasing, while the GN2 and GN4 results do not, following the recommendations of Section 7.2.2. The DN4 results without and with de-biasing are shown in Figure 3 to demonstrate the advantages of de-biasing when it works.

Note that the StOMP results were generated with the maximum number of stages set to 30, FAR thresholding, and $\alpha = 0.015$. The strange island shapes in StOMP's DN2 and DN4 results (also present for DN3, although this case is not shown) seem to be attributable to StOMP reaching the maximum number of stages. This serves to highlight some of the difficulties of working with StOMP—while the case of (scaled) Gaussian $A$ and $\sigma_1 = \sigma_2 = 0$ is well-characterized in [15], the authors do not clearly describe how to set the thresholding parameters and number of stages when different $A$ are used or noise is added to the system. We were obviously able to obtain good results for the scenarios listed in Table 7.3, but this required some trial and error on our part. For instance, we found it necessary to deviate from the default settings of 10 stages, FDR thresholding and $q = 0.5$.

The designers of StOMP claim some robustness to measurement noise in [15], but do not discuss this situation in great detail. They do show a phase plot for problems with scaled Gaussian A matrices and measurement noise for which the signal to noise ratio $\|x_s\|/\|A^T\epsilon_2\|$ is equal to 10. By generalizing the signal to noise ratio to $\|x_s\|/\|\epsilon_1 + A^T\epsilon_2\|$, we report average signal to noise ratios for the cases shown in Figure 2 in Table 3.

Table 3: Average signal-to-noise ratios ±2 standard deviations for phase plots in Figure 2.

| | *N1 | *N2 | *N4 |
|---|---|---|---|
| G* | 9E7 ± 1E7 | 180 ± 40 | 90 ± 20 |
| D* | 1E8 ± 4E6 | 180 ± 40 | 90 ± 20 |

Finally, we emphasize FPC's ability to recover a sparse signal in the presence of noise in Figure 4, which depicts $x_s$, $x_s + \epsilon_1$ and the recovered signal $x$ (without de-biasing) for a small

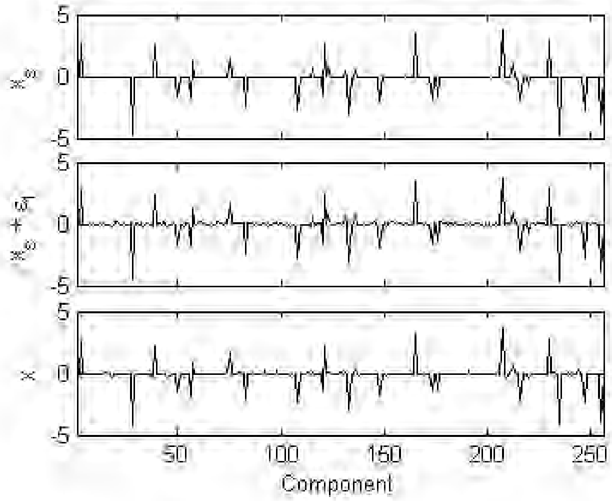

Figure 4: The original, noisy and recovered signals for a compressed sensing problem with a partial DCT $A$ matrix of size $128 \times 256$, $k = 32$, and $\sigma_1 = \sigma_2 = 0.1$.

problem ($n = 256$, $m = 128$, $\rho = 32$) with DCT $A$, $\sigma_1 = 0.1$ and $\sigma_2 = 0.1$, which corresponds to a signal to noise ratio of 4.4.

### 7.5 Sensitivity to $\mu$: A Comparison with GPSR and l1_ls

Intuitively we know that $\bar{\mu}$ should be inversely proportional to $\|Ax_s - b\|$, a fact that is reflected in the recommendations of Section 7.1. But is the recommended value of $\bar{\mu}$ overly conservative, how sensitive is the $\ell_1$-regularized problem (63) to $\mu$, and how does this sensitivity translate into the accuracy required of the $\sigma_1$ and $\sigma_2$ estimates? Furthermore, do FPC, GPSR-BB and l1_ls perform well for all $\mu$ values of interest?

To answer these questions, we solved randomly generated compressed sensing problems for a wide range of $\mu$ values. For simplicity we set $n = 1024$, $m = 512$, and varied the sparsity level among $\rho = 0.01$ ($k = 5$), 0.1 (51) and 0.2 (102). Experiments were conducted for every combination of algorithm (FPC, GPSR, l1_ls) and noise scenario (Table 7.3). All data points represent an average over 10 problems.

A representative sample of the experimental results is shown in Figures 5 and 6, which depict relative 2-norm error and CPU times for all algorithms and noise scenarios as a function of $\mu$. Figure 5 does this for Gaussian $A$ matrices with $\rho = 0.2$; Figure 6 shows results for partial DCT matrices and $\rho = 0.1$. Note that these results do not include any accuracy gains from de-biasing, nor do the CPU times include anything other than solving (63).

Several conclusions may be drawn from this data. First of all, l1_ls is robust to $\mu$ and produces accurate reconstructions, but is significantly slower than both FPC and GPSR-BB. As shown in the next section, this conclusion holds for all values of $n$.

Second, GPSR-BB is not robust to $\mu$. Invariably, as $\mu$ is increased one reaches a point where GPSR-BB fails, that is, GPSR-BB is no longer able to recover the original signal. Since FPC and l1_ls do recover $x_s$ to a reasonable accuracy at these $\mu$ values, we must conclude that GPSR-BB is no longer converging to a solution of (63). Although not depicted here, the full results clearly show that GPSR-BB is able to tolerate larger $\mu$ values when $x_s$ is sparser ($\rho$ is smaller).

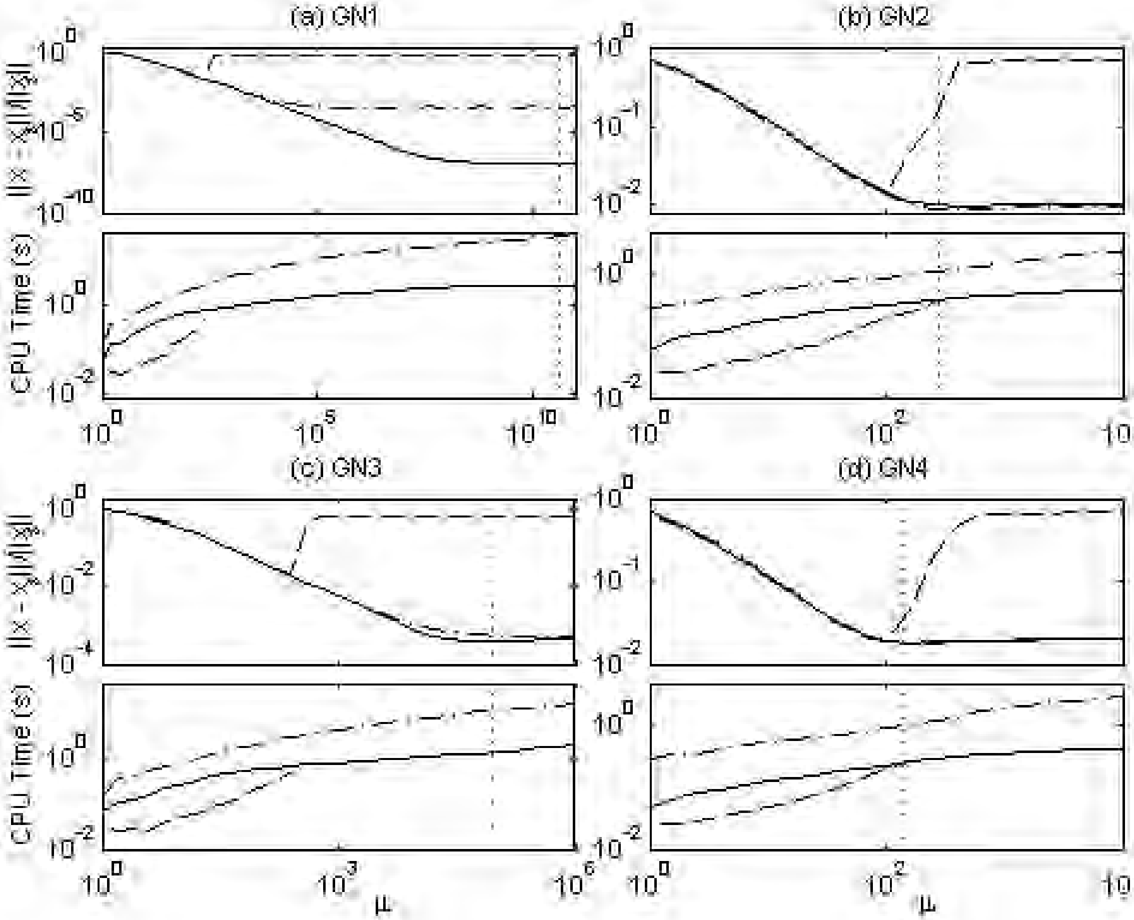

Figure 5: Relative recovery error and CPU times versus $\mu$ for Gaussian $A$ and $\rho = 0.2$. Each data point is the average over 10 compressed sensing problems solved with FPC (solid lines), GPSR-BB (dashed lines) or l1_ls (dash-dot lines). The vertical dotted lines represent the $\mu$ values recommended in Section 7.1 (computed with true noise levels and $\alpha = 0.5$). GPSR-BB's CPU times are not shown for $\mu$ values greater than that at which GPSR-BB's accuracy curve degrades from those of FPC and l1_ls.

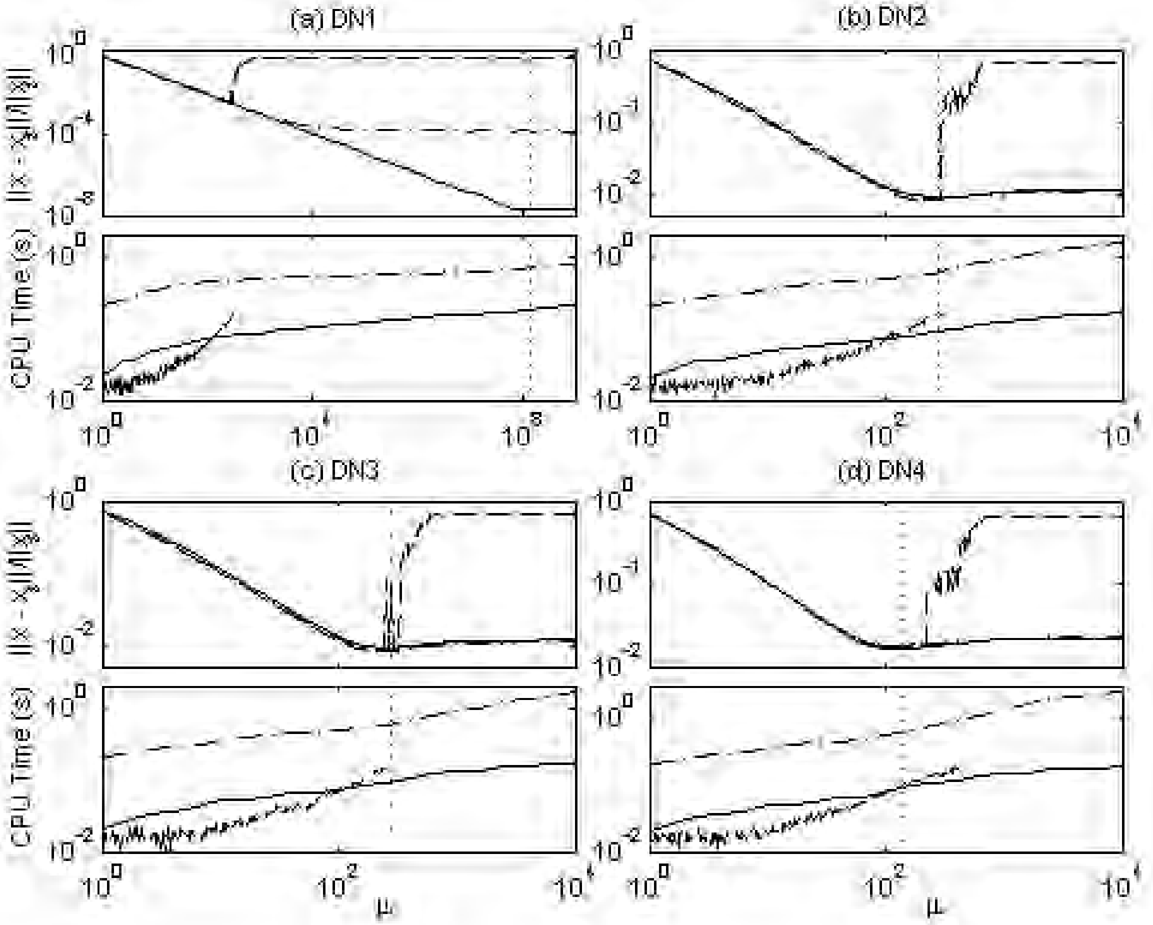

Figure 6: Relative recovery error and CPU times versus $\mu$ for DCT $A$ and $\rho = 0.1$. Each data point is the average over 10 compressed sensing problems solved with FPC (solid lines), GPSR-BB (dashed lines) or l1_ls (dash-dot lines). The vertical dotted lines represent the $\mu$ values recommended in Section 7.1 (computed with true noise levels and $\alpha = 0.5$). GPSR-BB's CPU times are not shown for $\mu$ values greater than that at which GPSR-BB's accuracy curve degrades from those of FPC and l1_ls.

The results also support the values of $M$ and $\mu$ specified in Section 7.1, as the dotted lines, which represent the recommended $\mu$ values, generally lie close to the smallest $\mu$ value for which maximum accuracy is attained. Of course, these recommended $M$ and $\mu$ values made use of the true noise levels $\sigma_1$ and $\sigma_2$, which are not generally known, and a critical $\chi^2$ value, which depends on the parameter $\alpha$. It turns out that $\bar{\mu}$ is not very sensitive to $\alpha$. For example, when $n = 1024$ and $m = 512$, $\sqrt{n/\chi^2_{1-\alpha,m}} = 1.34, 1.38, 1.42, 1.44,$ and $1.49$ when $\alpha = 0.05, 0.25, 0.5, 0.75$ and $0.95$, respectively, such that any $\alpha \in [0.05, 0.95]$ produces comparable $\bar{\mu}$ values. With regards to noise levels, Figures 5 and 6 indicate that order of magnitude estimates are sufficient for achieving good accuracy in a reasonable amount of time. In general one should err on the side of larger $\mu$ (smaller noise estimates) as this ensures accurate recovery. The only exception to this guideline is very sparse problems. Our results for $\rho = 0.01$ indicate that the relative error versus $\mu$ curve has a more distinct minimum, and that the recommended $\mu$ value tends to overestimate that minimum, when the original signal is very sparse. However, even in this case an order of magnitude estimate will give reasonable results.

Finally, note that the FPC and GPSR-BB solution times are quite comparable at $\mu$ values for which the original signal is recovered and GPSR-BB is able to converge to a solution of (63). In particular, whether FPC or GPSR-BB is faster is both problem-specific and depends on $\mu$. Thus, based purely on timing, it would be hard to choose between the two methods. However, GPSR-BB often fails at $\mu$ values smaller than what is necessary for accurate recovery. As mentioned in the Introduction, we suspect that this problem has its origins with the use of Barzilai-Borwein steps in a constrained setting.

Before discussing the next set of experiments, we take this opportunity to support the de-biasing recommendations of Section 7.2.2. Figure 7 shows FPC results with and without de-biasing for two cases: DN1 and DN2 with $\rho = 0.2$. In the first case, when there is no signal noise and little measurement noise ($\sigma_2$ =1E-8), post-processing with Algorithm 2 is inferior the FPC results at the recommended $\mu$ value because of the iterative solver tolerance used in the de-biasing algorithm. However, running FPC with $\bar{\mu} = 5000$ and then de-biasing provides acceptable results (relative error of 7E-6) in significantly less time as compared to running FPC with $\bar{\mu}$ =1E8. The DN2 results are representative of de-biasing in the case of DCT $A$ and noisy measurements: de-biasing provides more accuracy at low cost. In contrast, de-biasing performs well for Gaussian $A$ when $\sigma_1 = 0$, but degrades the results when $\sigma_1 \neq 0$ and $n$ is moderate to large.

## 7.6 Computational Time Comparison

In this section we investigate the performance of FPC, StOMP, GPSR-BB and l1_ls as a function of problem size. The experimental set-up is identical to that in the previous section, except that $\mu$ is fixed, $n$ is incrementally increased, and each data point is the average of five runs rather than 10.

The FPC and StOMP results for Gaussian $A$ matrices, noise scenarios two and three, and $\rho = k/m = 0.2$ and $0.1$ are shown in Table 4. ($\delta = m/n = 0.5$ in all cases.) The non-parenthetical numbers in the body of Table 4 are CPU times, either the time it took to generate the problem data (data), or to solve the problem with the given method (solve). The parenthetical numbers are the corresponding CPU times divided by the minimum CPU time, where the minimum is taken over the two methods. For instance, for GN2, $n = 512$ and $\rho = 0.2$, the StOMP problem data is generated faster than FPC's so that the parenthetical numbers

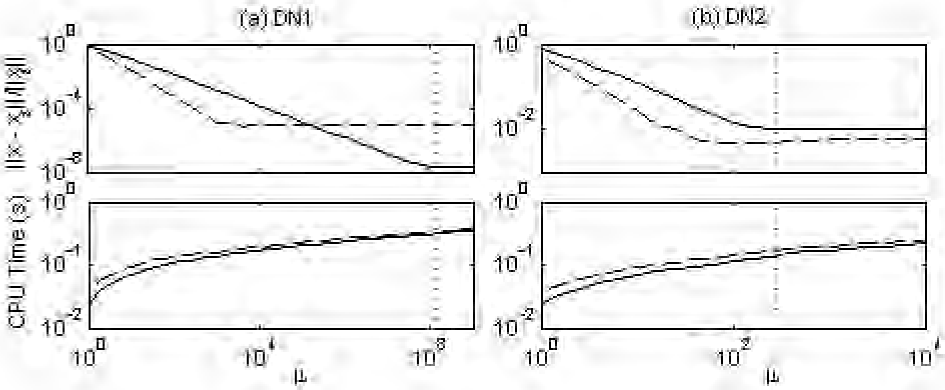

Figure 7: Relative recovery error and CPU times versus $\mu$ for DCT $A$ and $\rho = 0.2$. Each data point is the average over 10 compressed sensing problems solved with FPC without (solid lines) and with (dashed lines) de-biasing. The vertical dotted lines represent the $\mu$ values recommended in Section 7.1.

for the data CPU times are 1.17/0.014 = (84) for FPC and .014/.014 = (1) for StOMP, but FPC's solve time is faster, so the parenthetical numbers in this case are 0.050/0.050 = (1) for FPC and 0.119/0.050 = (2.4) for StOMP. The numbers below the method names are average relative 2-norm errors plus or minus one standard deviation. Note that the relative error is generally independent of problem size.

Following our previous discussions, the FPC results use the recommended $M$ and $\mu$ values; the GN3 ($\sigma_1 = 0$ and $\sigma_2$ =5E-3) results include de-biasing, but the GN2 ($\sigma_1$ = 5E-3 and $\sigma_2$ =1E-3) results do not. The data times for FPC include that for generating $A =$ `randn(m,n)` and calculating $M$ and $\mu$. StOMP's data times include generating $A =$ `randn(m,n)` and scaling $A$'s columns to unit norm.

The entirety of the Gaussian data (all noise scenarios and sparsity levels) reveals that both algorithms behave differently depending on whether or not there is noise on the original signal. As expected, it takes much longer to generate the FPC problem data when $\sigma_1 \neq 0$, because in this case $M$ and the extreme eigenvalues of $M^{1/2}AA^TM^{1/2}$ must be calculated outright. However, the solve times in this case are actually faster than when the overall noise level is similar but $\sigma_1 = 0$, which is evident when one compares the GN2 and GN3 results in Table 4. In particular, the GN2 solve times tend to be 3-7 times faster than the corresponding GN3 solve times.

StOMP, on the other hand, has fairly constant data generation times, but takes longer to solve problems for which $\sigma_1 \neq 0$. Again, it is best to compare GN2 and GN3; doing so reveals the GN3 solve times to be about twice as fast as the corresponding GN2 solve times.

For the noise-free signal cases (GN1 and GN3), FPC always has the fastest data-generation times, and has relatively better solution times for larger, sparser problems, as compared to StOMP. FPC boasts the fastest solution times for the largest problems reported in each case, except for GN1 and $\rho = 0.2$. When there is noise on the original signal (GN2 and GN4), FPC always has the fastest solve times (up to 28 times faster in the most favorable cases), but has very slow data-generation times, especially for larger $n$. An alternative formulation that addresses this issue is investigated below.

Continuing with the Gaussian scenarios, we ran analogous time trials for l1_ls and GPSR-BB with the recommended $M$ and $\mu$ values, when possible. Based on our $\mu$ study results, l1_ls

Table 4: Timing study results for Gaussian A matrices. The FPC GN3 accuracy and solve time results include de-biasing, but the GN2 results do not. For purposes of comparison, note that de-biasing never represents more than three percent of the reported solve time.

| **GN2** | $\rho = 0.2$ | | | | $\rho = 0.1$ | | | |
|---|---|---|---|---|---|---|---|---|
| | FPC | | StOMP | | FPC | | StOMP | |
| | 9.2E-3 ± 2.1E-4 | | 9.9E-3 ± 5.2E-4 | | 8.9E-3 ± 3.4E-4 | | 1.2E-2 ± 5.5E-4 | |
| n | data | solve | data | solve | data | solve | data | solve |
| 512 | 1.17 (84) | 0.089 (1) | 0.014 (1) | 0.119 (1.3) | 1.17 (83) | 0.057 (1) | 0.014 (1) | 0.108 (1.9) |
| 1024 | 11.5 (240) | 0.267 (1) | 0.049 (1) | 0.593 (2.2) | 11.9 (240) | 0.178 (1) | 0.048 (1) | 0.618 (3.5) |
| 2048 | 95 (510) | 0.916 (1) | 0.185 (1) | 8.59 (9.4) | 94.5 (510) | 0.620 (1) | 0.185 (1) | 4.99 (8.0) |
| 4096 | 772 (1000) | 3.88 (1) | 0.730 (1) | 71.6 (18) | 747 (1000) | 2.32 (1) | 0.730 (1) | 48.3 (21) |
| **GN3** | $\rho = 0.2$ | | | | $\rho = 0.1$ | | | |
| | FPC | | StOMP | | FPC | | StOMP | |
| | 3.0E-4 ± 1.4E-4 | | 6.5E-3 ± 6.1E-4 | | 3.9E-4 ± 2.1E-4 | | 7.4E-3 ± 6.1E-4 | |
| n | data | solve | data | solve | data | solve | data | solve |
| 512 | 0.010 (1) | 0.395 (5.3) | 0.013 (1.3) | 0.075 (1) | 0.010 (1) | 0.166 (3.2) | 0.014 (1.3) | 0.052 (1) |
| 1024 | 0.030 (1) | 1.28 (2.9) | 0.049 (1.6) | 0.447 (1) | 0.030 (1) | 0.674 (2.3) | 0.049 (1.6) | 0.298 (1) |
| 2048 | 0.110 (1) | 4.84 (1.3) | 0.185 (1.7) | 3.80 (1) | 0.110 (1) | 2.29 (1) | 0.185 (1.7) | 2.92 (1.3) |
| 4096 | 0.428 (1) | 19.1 (1) | 0.797 (1.9) | 31.6 (1.7) | 0.427 (1) | 9.16 (1) | 0.761 (1.8) | 19.3 (2.1) |

was tested for all scenarios, but GPSR-BB was limited to noise scenario, $\rho$ combinations of (GN2, 0.01), (GN4, 0.1) and (GN4,0.01), which are generally characterized by noisy, sparse signals. Note that, following Section 7.3, l1_ls and GPSR-BB have the same data-generation times as FPC and give similar accuracy results before de-biasing.

Overall, l1_ls is much slower than FPC in all cases. Its solve times are sometimes faster than StOMP's, in particular when there is noise on the original signal and the problem is large. The best ratio of an l1_ls solve time to the best solve time between FPC and StOMP is 3.6 (for GN2, $\rho = 0.2$ and $n = 4096$). A ratio of 10-20 is more typical; and the worst is 1040 (for GN1, $\rho = 0.01$ and $n = 512$).

As stated in Section 7.5, GPSR-BB has solve times comparable to those of FPC when it works. For the Gaussian $A$ cases mentioned above, GPSR-BB is about 50% slower than FPC for the second noise scenario (and $\rho = 0.01$), and is 15-50% slower in the fourth noise scenario, achieving its best relative solve times when the original signal is large and sparse ($n = 4096$ and $\rho = 0.01$).

One question raised by this data is whether or not reasonable results can be obtained for the noisy signal scenarios (GN2 and GN4) if $M$ is set to an appropriate multiple of $I$, rather than to $(\sigma_1^2 AA^T + \sigma_2^2 I)^{-1}$, which is itself expensive to calculate, and also gives rise to a $\bar{\mu}$ estimate that contains the largest and smallest eigenvalues of $M^{1/2}AA^TM^{1/2}$. To this end we repeated our GN2 and GN4 timing runs with

$$M = (\sigma_1^2 \overline{\sigma}^2 I + \sigma_2^2 I)^{-1}, \tag{74}$$

where $\overline{\sigma}^2 = \left(1 + \sqrt{\frac{m}{n}}\right)^2 n$ is an upper bound approximation to $\lambda_{\max}(AA^T)$. Following the developments of Section 7.1, the corresponding $\bar{\mu}$ estimate is

$$\bar{\mu} = \frac{\overline{\sigma}^2}{\underline{\sigma}\sqrt{\sigma_1^2\overline{\sigma}^2 + \sigma_2^2}}\sqrt{\frac{n}{\chi^2_{1-\alpha,m}}}, \tag{75}$$

Table 5: Timing study results for Scenario GN4 and $\rho = 0.1$. Comparison of FPC with $M$ as in (66), FPC with an approximate $M$ (74), and StOMP.

| | FPC<br>1.7E-2 ± 7.4E-4 | | FPC Approx.<br>1.9E-2 ± 6.2E-4 | | StOMP<br>2.5E-2 ± 6.4E-4 | |
|---|---|---|---|---|---|---|
| n | data | solve | data | solve | data | solve |
| 512 | 1.16 (120) | 0.059 (1) | 0.010 (1) | 0.148 (2.5) | 0.013 (1.3) | 0.110 (1.9) |
| 1024 | 11.6 (380) | 0.180 (1) | 0.030 (1) | 0.459 (2.6) | 0.048 (1.6) | 0.518 (2.9) |
| 2048 | 94.5 (860) | 0.614 (1) | 0.109 (1) | 1.55 (2.5) | 0.185 (1.7) | 6.67 (11) |
| 4096 | 742 (1700) | 2.37 (1) | 0.429 (1) | 6.19 (2.6) | 0.729 (1.7) | 49.4 (21) |
| 8192 | - | - | 1.71 (1) | 27.2 (1) | 2.87 (1.7) | 358 (13) |

which matches (70) in the case that $\sigma_1 = 0$. Relevant timing results for GN4, $\rho = 0.1$, are shown in Table 5.

The data show that the approximate $M$ largely accomplishes all we could hope for. While the FPC solve times are about two times slower with the approximate, rather than the full, $M$ matrix, the data times are hundreds of times faster, and FPC retains its advantage over StOMP. Thus we recommend using the full $M$ matrix when many problems of the same size and noise characteristics are to be solved, and an approximate $M$ when any of $m$, $n$, $\sigma_1$ or $\sigma_2$ vary.

Tables 6 and 7 list FPC, StOMP and GPSR-BB data for two noise scenarios with DCT $A$ . In this case there is no weighting matrix $M$, and the problems solved by each algorithm (including StOMP) are completely equivalent. Therefore, we do not show the data-generation times, which were about two orders of magnitude smaller than the solution times. The third and fourth noise scenarios' data are not shown because the trends in those data are very similar to those seen in DN2. De-biasing was applied to FPC and GPSR-BB in all cases; for DN1, $\bar{\mu} = 5000$. GPSR-BB results are only shown when GPSR-BB can successfully solve (63) with the same $\bar{\mu}$ as FPC.

FPC and StOMP have comparable solution times in the low-noise DN1 scenarios, with FPC preferred when the original signal is large and less sparse. StOMP does best when the original signals are very sparse, and preferably small, both in the DN1 case, and with noisy measurements, as in DN2, DN3 and DN4. In all cases FPC is relatively faster, as compared to StOMP, with increasing $n$. While FPC never overtakes StOMP with DN1 and $\rho = 0.01$ for the problem sizes shown, it is apparent that it would do so at larger values of $n$; for the remainder of the noise scenarios, FPC was faster than StOMP for all problems with $n = 1048576$ or greater, often surpassing StOMP for smaller $n$ when $\rho = 0.1$ or 0.2.

This section supports the observation in Section 7.5 that FPC and GPSR-BB solve times are generally comparable when GPSR-BB is able to recover the original signal. Table 7 contains a sampling of the relevant data; comparable GPSR-BB data was also generated for the pairs (DN3, $\rho = 0.01$), (DN4, $\rho = 0.1$), and (DN4, $\rho = 0.01$). For large $n$, FPC was faster than GPSR-BB in the (DN3, $\rho = 0.01$) and (DN4, $\rho = 0.1$) cases, and GPSR-BB was faster in the (DN4, $\rho = 0.01$) case; the two algorithms never differed by more than 30%. As in Section 7.5, l1_ls proved to be slower than the other algorithms; it was typically three to ten times slower than the fastest algorithm.

Table 6: Timing study results for Scenario DN1 and all sparsity levels. The FPC accuracy and solve time results were computed with $\bar{\mu} = 5000$ and de-biasing.

| | $\rho = 0.2$ | | $\rho = 0.1$ | | $\rho = 0.01$ | |
|---|---|---|---|---|---|---|
| | FPC | StOMP | FPC | StOMP | FPC | StOMP |
| n | 7.5E-6 ± 9.7E-7 | 6.2E-6 ± 1.2E-6 | 7.1E-6 ± 1.1E-6 | 7.3E-6 ± 1.5E-6 | 2.2E-6 ± 1.4E-6 | 1.7E-6 ± 7.5E-7 |
| 512 | 0.106 (1) | 0.106 (1) | 0.070 (1.2) | 0.057 (1) | 0.036 (3.0) | 0.012 (1) |
| 1024 | 0.192 (1) | 0.193 (1.0) | 0.111 (1.1) | 0.104 (1) | 0.064 (2.9) | 0.022 (1) |
| 2048 | 0.475 (1.3) | 0.357 (1) | 0.223 (1.0) | 0.221 (1) | 0.131 (2.1) | 0.063 (1) |
| 4096 | 0.724 (1) | 0.801 (1.1) | 0.422 (1) | 0.449 (1.1) | 0.254 (2.1) | 0.123 (1) |
| 8192 | 1.37 (1) | 1.56 (1.1) | 0.845 (1) | 0.879 (1.0) | 0.482 (1.6) | 0.305 (1) |
| 16384 | 2.56 (1) | 2.97 (1.2) | 1.58 (1) | 1.81 (1.2) | 0.949 (1.8) | 0.538 (1) |
| 32768 | 5.32 (1) | 6.66 (1.2) | 3.29 (1) | 3.64 (1.1) | 2.06 (1.8) | 1.14 (1) |
| 65536 | 12.6 (1) | 14.2 (1.1) | 7.94 (1) | 8.11 (1.0) | 4.71 (1.9) | 2.52 (1) |
| 131072 | 26.2 (1) | 29.0 (1.1) | 16.4 (1.0) | 15.8 (1) | 9.59 (1.4) | 7.02 (1) |
| 262144 | 53.1 (1) | 64.2 (1.2) | 33.2 (1) | 33.2 (1) | 19.8 (1.6) | 12.4 (1) |
| 524288 | 107 (1) | 119 (1.1) | 66.9 (1.0) | 64.4 (1) | 40.4 (1.3) | 30.2 (1) |
| 1048576 | 227 (1) | 257 (1.1) | 140 (1) | 141 (1.0) | 84.5 (1.3) | 63.3 (1) |
| 2097152 | 450 (1) | 511 (1.1) | 281 (1.0) | 268 (1) | 168 (1.3) | 128 (1) |

Table 7: Timing study results for Scenario DN2 and all sparsity levels. The FPC and GPSR accuracy and solve time results include de-biasing.

| | $\rho = 0.2$ | | $\rho = 0.1$ | | $\rho = 0.01$ | | |
|---|---|---|---|---|---|---|---|
| | FPC | StOMP | FPC | StOMP | FPC | GPSR | StOMP |
| n | 4.6E-3 ± 8.0E-5 | 4.3E-3 ± 1.8E-4 | 4.3E-3 ± 8.4E-5 | 4.2E-3 ± 1.4E-4 | 6.4E-3 ± 7.3E-4 | 3.3E-2 ± 3.2E-3 | 5.9E-3 ± 1.0E-3 |
| 512 | 0.093 (1) | 0.109 (1.2) | 0.068 (1.2) | 0.055 (1) | 0.041 (2.7) | 0.053 (3.4) | 0.015 (1) |
| 1024 | 0.158 (1) | 0.189 (1.2) | 0.112 (1.1) | 0.100 (1) | 0.076 (2.8) | 0.083 (3.1) | 0.027 (1) |
| 2048 | 0.345 (1) | 0.443 (1.3) | 0.233 (1) | 0.283 (1.2) | 0.158 (2.0) | 0.190 (2.4) | 0.078 (1) |
| 4096 | 0.649 (1) | 0.837 (1.3) | 0.454 (1) | 0.518 (1.1) | 0.308 (2.0) | 0.369 (2.4) | 0.156 (1) |
| 8192 | 1.28 (1) | 1.78 (1.4) | 0.867 (1) | 0.990 (1.1) | 0.590 (1.4) | 0.681 (1.6) | 0.421 (1) |
| 16384 | 2.54 (1) | 3.45 (1.4) | 1.70 (1) | 2.15 (1.3) | 1.17 (1.4) | 1.27 (1.5) | 0.865 (1) |
| 32768 | 5.03 (1) | 7.24 (1.4) | 3.56 (1) | 4.38 (1.2) | 2.35 (1.3) | 2.68 (1.5) | 1.77 (1) |
| 65536 | 11.2 (1) | 19.9 (1.8) | 7.50 (1) | 11.2 (1.5) | 5.07 (1.1) | 6.54 (1.4) | 4.58 (1) |
| 131072 | 23.2 (1) | 40.7 (1.8) | 16.2 (1) | 27.0 (1.7) | 10.6 (1.0) | 13.7 (1.3) | 10.5 (1) |
| 262144 | 46.8 (1) | 91.8 (2.0) | 32.4 (1) | 65.9 (2.0) | 20.9 (1) | 27.5 (1.3) | 25.6 (1.2) |
| 524288 | 95.2 (1) | 199 (2.1) | 65.1 (1) | 128 (2.0) | 45.1 (1) | 50.4 (1.1) | 56.1 (1.2) |
| 1048576 | 199 (1) | 466 (2.4) | 138 (1) | 298 (2.2) | 90.3 (1) | 103 (1.1) | 145 (1.6) |
| 2097152 | 397 (1) | - | 274 (1) | 685 (2.5) | 197 (1) | 206 (1.0) | 327 (1.7) |

### 7.7 Summary of Numerical Results

FPC, StOMP and l1_ls all give accurate reconstruction results for a wide variety of compressed sensing problems, even in the face of noisy signals and measurements. For easy reconstructions, any of these algorithms should be sufficient with regards to accuracy, however, the phase transitions for StOMP differ from those of FPC (which are the same as l1_ls's since they are both based on the formulation (63)), and so phase plots should be consulted in borderline cases. In general, noise reduces StOMP's "good" region (with regards to acceptable $m/n$ and $k/m$) far more than FPC's.

The current version of GPSR-BB, on the other hand, is not robust with respect to the parameter $\bar{\mu}$ in the sense that for any given problem, there is some $\mu$ value above which GPSR-BB no longer converges to a solution of (63). In many cases, this cut-off value is not sufficiently large to produce an accurate compressed sensing reconstruction. GPSR-BB is most likely to perform as well as FPC and l1_ls when the original signal is very sparse, and the noise levels are high.

With regards to speed, l1_ls is nearly always the slowest algorithm of the four. For the cases studied in this work, FPC will always be faster than StOMP for large enough problems. For Gaussian $A$ matrices and $\sigma_1 = 0$ (no signal noise), StOMP performs best for smaller, less sparse signals. StOMP is considerably slower than FPC when there is signal noise, even if an approximate $M$ matrix (a multiple of the identity) is used in order to avoid expensive data preparations. For partial DCT $A$ matrices, the trends for StOMP relative to FPC are similar (chiefly, the larger the problem size is, the slower is StOMP relative to FPC), except that in this case the ratios of StOMP time vs. FPC time are better when signals are sparser, and the presence or absence of signal noise longer significantly affects the running time of StOMP.

When GPSR-BB is able to solve (63) at the desired value of $\bar{\mu}$, its solve times are very comparable to FPC's. For the scenarios considered in this work, FPC was 15-70% faster than GPSR-BB when $A$ was a Gaussian matrix, but 25% slower to 40% faster than GPSR-BB when $A$ was a partial DCT matrix. (In the latter case, GPSR-BB was significantly faster than FPC only at the highest noise and sparsity levels.)

## 8 Conclusions

We investigated the use of the forward-backward splitting technique, combined with a continuation (path-following) strategy, for solving $\ell_1$-norm regularized convex optimization problems. Our theoretical analysis yields convergence results stronger than what could be obtained from applying existing general theory to our setting. In particular, we have established finite convergence for some quantities and a $q$-linear convergence rate without a strict convexity assumption. Interestingly, our rate of convergence results imply, in a general sense, that sparser solutions correspond to faster rates of convergence, which agrees with what has been observed in practice.

We conducted a comprehensive computational study comparing our fixed point continuation (FPC) algorithm with three state-of-the-art compressed sensing recovery algorithms. Extensive numerical results indicate that, overall, FPC outperforms the other algorithms, especially when the data are noisy and the problem is large. The strong performance of FPC as applied to compressed sensing lends support to our belief that the fixed point continuation framework should also yield robust and efficient algorithms for more general $\ell_1$-regularized optimization problems, which is a topic of further research.

## Acknowledgements

The work of E. Hale was supported by an NSF VIGRE grant (DSM 0240058). The work of W. Yin was supported by an internal faculty research grant from the Dean of Engineering at Rice University. The work of Y. Zhang was supported in part by NSF Grant DMS-0405831.

## References

[1] D. Baron, M. Wakin, M. Duarte, S. Sarvotham, and R. Baraniuk. Distributed compressed sensing. *Preprint*, 2005.

[2] E. Candès and J. Romberg. Quantitative robust uncertainty principles and optimally sparse decompositions. *Foundations of Computational Mathematics*, 6(2):227–254, 2006.

[3] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52:489–509, 2006.

[4] E. Candès and T. Tao. Near optimal signal recovery from random projections: universal encoding strategies. *IEEE Transactions on Information Theory*, 52(1):5406–5425, 2006.

[5] 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 Transactions on Image Processing*, 7(3):319–335, 1998.

[6] H.-G. Chen and R. T. Rockafellar. Convergence rates in forward-backward splitting. *SIAM Journal on Optimization*, 7:421–444, 1997.

[7] S. Chen, D. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

[8] J. Claerbout and F. Muir. Robust modelling of erratic data. *Geophysics*, 38:826–844, 1973.

[9] P. L. Combettes and J.-C. Pesquet. Proximal thresholding algorithm for minimization over orthonormal bases. *To appear in SIAM Journal on Optimization*, 2007.

[10] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications in Pure and Applied Mathematics*, 57:1413–1457, 2004.

[11] D. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[12] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52:1289–1306, 2006.

[13] D. Donoho and J. Tanner. Neighborliness of randomly-projected simplices in high dimensions. *Proc. National Academy of Sciences*, 102(27):9452–9457, 2005.

[14] D. Donoho and Y. Tsaig. Fast solutions of $\ell_1$-norm minimization problems when the solution may be sparse. *Technical report online.*, 2006.

[15] D. Donoho, Y. Tsaig, I. Drori, and J.-C. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. *Preprint*, 2006.

[16] J. Douglas and H. H. Rachford. On the numerical solution of the heat conduction problem in 2 and 3 space variables. *Transactions of the American Mathematical Society*, 82:421–439, 1956.

[17] J. Eckstein. *Splitting methods for monotone operators with applications to parallel optimization.* PhD thesis, Massachusetts Institute of Technology, 1989.

[18] A. Edelman. Eigenvalues and condition numbers of random matrices. *SIAM Journal on Matrix Analysis and Applications*, 9(4):543–560, 1988.

[19] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[20] 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.

[21] M. Figueiredo and R. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 12:906–916, 2003.

[22] 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.

[23] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *Preprint*, 2007.

[24] D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Solution of Boundary Value Problems.* Horth-Hollan, Amsterdam, 1983.

[25] D. Jonsson. Some limit theorems for the eigenvalues of a sample covariance matrix. *Jounral of Multivariate Analysis*, 12:1–38, 1982.

[26] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. A method for large-scale $l_1$-regularized least squares problems with applications in signal processing and statistics. *Manuscript*, 2007.

[27] 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.

[28] $l_1$–Magic. The $l_1$–Magic tool box. URL: http://www.acm.caltech.edu/l1magic, 2006.

[29] J. Laska, S. Kirolos, M. Duarte, T. Ragheb, R. Baraniuk, and Y. Massoud. Theory and implementaion of an analog-to information converter using random demodulation. In *Proceedings of the IEEE International Symposium on Circuites and Systems (ISCAS)*, New Orleans, Louisiana, 2007.

[30] J. Laska, S. Kirolos, Y. Massoud, R. Baraniuk, A. Gilbert, M. Iwen, and M. Strauss. Radnom sampling for analog-to-information conversion of wideband signals. In *Proceedings of the IEEE Dallas Circuits and Systems Workshop*, Dallas, Texas, 2006.

[31] S. Levy and P. Fullagar. Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution. *Geophysics*, 46:1235–1243, 1981.

[32] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16:964–979, 1979.

[33] Z.-Q. Luo and P. Tseng. On the linear convergence of descent methods for convex essentially smooth minimization. *SIAM Journal on Control and Optimization*, 30(2):408–425, 1990.

[34] M. Lustig, D. Donoho, and J. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Preprint*, 2007.

[35] D. Malioutov, M. Çetin, and A. Willsky. Homotopy continuation for sparse signal representation. In *Prceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 733–736, Philadelphia, PA, 2005.

[36] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[37] B. Mercier. Inéquations Variationnelles de la Mécanique. *Publications Mathématiques d'Orsay, Université de Paris-Sud, Orsay*, 80.01, 1980.

[38] A. Miller. *Subset Selection in Regression.* Chapman and Hall, 2002.

[39] M. A. Noor. Splitting methods for pseudomonotone mixed variational inequalites. *Journal of Mathematical Analysis and Applications*, 246:174–188, 2000.

[40] M. Osborne, B. Presnell, and B. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20:389–403, 2000.

[41] J.-S. Pang. A posteriori error bounds for the linearly-constrained variational inequality problem. *Mathematical Methods of Operations Research*, 12:474–484, 1987.

[42] G. B. Passty. Ergodic convergence to a zero of the sum of monotone operators in Hilbert space. *Journal of Mathematical Analysis and Applications*, 72:383–390, 1979.

[43] D. H. Peaceman and H. H. Rachford. The numerical solution of parabolic elliptic differential equations. *SIAM Journal on Applied Mathematics*, 3:28–41, 1955.

[44] R. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, 1970.

[45] M. Rudelson and R. Vershynin. Geometric approach to error-correcting codes and reconstruction of signals. *Int. Math. Res. Not.*, (64):4019–4041, 2005.

[46] F. Santosa and W. Symes. Linear inversion of band-limited reflection histograms. *SIAM Journal on Scientific and Statistical Computing*, 7:1307–1330, 1986.

[47] 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.

[48] H. Taylor, S. Bank, and J. McCoy. Deconvolution with the $l_1$ norm. *Geophysics*, 44:39–52, 1979.

[49] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal Royal Statistical Society B*, 58:267–288, 1996.

[50] J. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2342, 2006.

[51] J. Tropp. Just relax: Convex programming methods for identifying sparse signals. *IEEE Transactions on Information Theory*, 51:1030–1051, 2006.

[52] 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.

[53] Y. Tsaig and D. Donoho. Extensions of compressed sesning. *Signal Processing*, 86(3):533–548, 2005.

[54] P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. *SIAM Journal on Control and Optimization*, 38(2):431–446, 2000.

[55] B. Turlach. On algorithms for solving least squares problems under an $L_1$ penalty or an $L_1$ constraint. In *Proceedings of the American Statistical Association; Statistical Computing Section*, pages 2572–2577, Alexandria, VA, 2005.

[56] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk. An architecture for compressiving image. In *Proceedings of the International Conference on Image Processing (ICIP)*, Atlanta, Georgia, 2006.

[57] 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.

[58] Y. Zhang. When is missing data recoverable? *Rice University CAAM Technical Report TR06-15*, 2006.

# A Appendix

### Proof of Lemma 3.2

*Proof.* For ease of notation, we drop the subscript $i$ by letting $p^1 = y_i^1, p^2 = y_i^2$.

First, when $\mathrm{sgn}(p^1) = \mathrm{sgn}(p^2) \neq 0$,

$$\begin{aligned}
& |\mathrm{sgn}(p^1)\max\{|p^1| - \nu, 0\} - \mathrm{sgn}(p^2)\max\{|p^2| - \nu, 0\}| \\
= \; & |\max\{|p^1| - \nu, 0\} - \max\{|p^2| - \nu, 0\}| \\
= \; & \begin{cases} \left||p^1| - |p^2|\right|, & |p^1| \geq \nu,\ |p^2| \geq \nu \\ \left||p^2| - \nu\right|, & |p^1| < \nu,\ |p^2| \geq \nu \\ \left||p^1| - \nu\right|, & |p^1| \geq \nu,\ |p^2| < \nu \\ 0, & |p^1| < \nu,\ |p^2| < \nu \end{cases} \\
\leq \; & \left||p^1| - |p^2|\right| \\
= \; & |p^1 - p^2|,
\end{aligned}$$

where the inequality holds as an equality if and only if both $|p^1| \geq \nu$ and $|p^2| \geq \nu$.

Second, when $\mathrm{sgn}(p^1) = \mathrm{sgn}(p^2) = 0$, (22) holds as an equality trivially for $p^1$ and $p^2$.

Third, when $\mathrm{sgn}(p^1) \neq \mathrm{sgn}(p^2)$ and both nonzero,

$$\begin{aligned}
& & & |\mathrm{sgn}(p^1)\max\{|p^1| - \nu, 0\} - \mathrm{sgn}(p^2)\max\{|p^2| - \nu, 0\}| \\
(\because \mathrm{sgn}(p^1) = -\mathrm{sgn}(p^2)) & & = \; & |\max\{|p^1| - \nu, 0\} + \max\{|p^2| - \nu, 0\}| \\
& & = \; & |\max\{|p^1| + |p^2| - 2\nu, |p^1| - |p^2|, |p^2| - |p^1|\}| \\
(\because \mathrm{sgn}(p^1) = -\mathrm{sgn}(p^2)) & & \leq \; & \left|\max\{|p^1 - p^2| - 2\nu, |p^1 + p^2|\}\right| \\
& & < \; & |p^1 - p^2|.
\end{aligned}$$

Last, when one and only one of $\mathrm{sgn}(p^1)$ and $\mathrm{sgn}(p^2)$ vanishes,

$$|\mathrm{sgn}(p^1)\max\{|p^1| - \nu, 0\} - \mathrm{sgn}(p^2)\max\{|p^2| - \nu, 0\}| < |p_1 - p_2|$$

follows from $|\mathrm{sgn}(p)\max\{|p| - \nu, 0\}| < |p|$ for $p \neq 0$. Therefore, we have proved the component-wise non-expansiveness of $s_\nu(\cdot)$.

Next, we prove Parts 1 through 6 one by one.

i. The first result, $\mathrm{sgn}(p^1) = \mathrm{sgn}(p^2)$, directly follows from (23) and the fact that whenever $\mathrm{sgn}(p^1) \neq \mathrm{sgn}(p^2)$, (22) holds with strictly inequality, which has been shown above.

   To obtain the second result, $s_\nu(p^1) - s_\nu(p^2) = p^1 - p^2$, we consider two cases: $p^1 = p^2$ and $p^1 \neq p^2$.

   If $p^1 = p^2$ then $s_\nu(p^1) - s_\nu(p^2) = p^1 - p^2 = 0$ holds obviously.

   If $p^1 \neq p^2$, we assume $p^1 > p^2$ without loss of generality. Under this assumption, we shall show $s_\nu(p^1) \geq s_\nu(p^2)$. In fact, it is trivial to verify from the definition of $s_\nu$ that in all of the following cases: $0 \geq p^1 > p^2$, $p^1 \geq 0 > p^2$, $p^1 > 0 \geq p^2$, and $p^1 > p^2 \geq 0$, we have $s_\nu(p^1) \geq s_\nu(p^2)$.

   Therefore, (23) gives $s_\nu(p^1) - s_\nu(p^2) = p^1 - p^2$.

   We state an additional result for the rest of proof below: whenever (23) holds (hence, $\mathrm{sgn}(p^1) = \mathrm{sgn}(p^2)$), we have

$$s_\nu(p^1) - s_\nu(p^2) = p^1 - p^2 + \mathrm{sgn}(p^1)([\nu - |p^1|]^+ - [\nu - |p^2|]^+) = p^1 - p^2, \tag{76}$$

   which follows from the expression (21) in Lemma 3.1.

ii. From (23), $\mathrm{sgn}(p^1) = \mathrm{sgn}(p^2)$, which cannot be zero (otherwise, $p^1 = p^2 = 0$ contradicts $p^1 \neq p^2$). Therefore, $[\nu - |p^1|]^+ = [\nu - |p^2|]^+$ follows from (76). Since $p^1 \neq p^2$ from (b), the only case that $[\nu - |p^1|]^+ = [\nu - |p^2|]^+$ can hold is $[\nu - |p^1|]^+ = [\nu - |p^2|]^+ = 0$; hence, $|p^1| \geq \nu$ and $|p^2| \geq \nu$. Therefore, the two conditions give $\nu \leq |p^1| \neq |p^2| \geq \nu$.

iii. If $p^2 = 0$, then $\mathrm{sgn}(p^1) = \mathrm{sgn}(p^2) = 0$, so $p^1 = 0$. Then, the result follows trivially.

If $p^2 \neq 0$, then $\mathrm{sgn}(p^1) = \mathrm{sgn}(p^2) \neq 0$, so (76) gives

$$[\nu - |p^1|]^+ = [\nu - |p^2|]^+. \tag{77}$$

Since $\nu - |p^2| > 0$, it follows from (77) that $\nu > |p^1|$ and $|p^1| = |p^2|$. Together with the fact that $\mathrm{sgn}(p^1) = \mathrm{sgn}(p^2)$, we have $p^1 = p^2$.

Altogether, we have $|p^1| = |p^2| < \nu$, $p^1 = p^2$, and hence $s_\nu(p^1) = s_\nu(p^2) = 0$ from Lemma 3.1.

iv. From (23) and $|p^2| > \nu$, $\mathrm{sgn}(p^1) = \mathrm{sgn}(p^2) \neq 0$ and thus holds (77) above. Also from $|p^2| > \nu$, $[\nu - |p^1|]^+ = [\nu - |p^2|]^+ = 0$; therefore, $|p^1| \geq \nu$.

v. If $p^1 = 0$, the result holds with equality trivially.

If $p^1 \neq 0$, we assume $p^1 > 0$ and $p^2 < 0$ without loss of generality. From Lemma 3.1 and because of $\mathrm{sgn}(p^1) > 0$ and $\mathrm{sgn}(p^1) < 0$, we have

$$\begin{aligned}
|s_\nu(p^1) - s_\nu(p^2)| &= s_\nu(p^1) - s_\nu(p^2) \\
&= \left(p^1 - \nu + [\nu - |p^1|]^+\right) - \left(p^2 + \nu - [\nu - |p^2|]^+\right) \\
&= \left(p^1 - \nu + [\nu - p^1]^+\right) - \left(p^2 + \nu\right) \\
&= p^1 - p^2 - 2\nu + [\nu - p^1]^+ \\
&\leq p^1 - p^2 - \nu \\
&= |p^1 - p^2| - \nu,
\end{aligned}$$

where the first and last equalities both follow from $p^1 > p^2$, and the inequality from $p^1 > 0$ and thus $[\nu - p^1]^+ \leq \nu$.

vi. The given conditions are equivalent to $|p^1| > \nu$ and $|p^2| \leq \nu$. Without loss of generality, we assume $p^1 > 0$; therefore, we have $p^1 > \nu \geq p^2$ and $s_\nu(p^1) > 0$ following from the assumption. Consequently, we get

$$\begin{aligned}
|s_\nu(p^1) - s_\nu(p^2)| &= s_\nu(p^1) \\
&= p^1 - \nu\mathrm{sgn}(p^1) + \mathrm{sgn}(p^1)[\nu - |p^1|]^+ \\
&= p^1 - \nu \\
&= p^1 - p^2 - (\nu - p^2) \\
&= |p^1 - p^2| - (\nu - p^2) \\
&\leq |p^1 - p^2| - (\nu - |p^2|).
\end{aligned}$$

It is easy to see that the above result holds with equality if $p^1 > \nu$, $p^2 \geq 0$ or $p^1 < -\nu$, $p^2 \leq 0$.

□