# Hierarchical Reconstruction of Sparse Signals

Ming Zhong*

Advisor: Dr. Eitan Tadmor†
*AMSC* and *CSCAMM*
University of Maryland College Park
College Park, Maryland 20742 USA

December 14, 2012

**Abstract**

We consider the constrained minimal $\ell_p$-norm problem: $\min_{\mathbf{x}\in\mathbf{R}^n}\{||\mathbf{x}||_{\ell_p} \mid A\mathbf{x} = A\mathbf{x}_*\}$, where $\mathbf{x}_*$ and $A$ are known. Such problem has been extensively studied in the Compressed Sensing Community and used to recover sparse signals. We go through the reasoning on why $p = 1$ would be the most suitable choice. Then we introduce Tikhonov Regularization to the aforementioned problem in order to avoid possible ill-posedness. We arrive at an unconstrained minimization problem: $\min_{\mathbf{x}\in\mathbf{R}^n}\{||\mathbf{x}||_{\ell_1} + \frac{\lambda}{2}||A\mathbf{x}_* - A\mathbf{x}||^2_{\ell_2}\}$. Through this regularized version, we reconstruct the approximation based on a given scale $\lambda$. Using the idea of multi-scale approximation, we adopt the method of Hierarchical Decomposition from Image Processing to resolve the sparse signals on layers of refined scales. We proceed to show that this Hierarchical Decomposition approach offers a systematic selection of the regularization/scale parameter $\lambda$. We also are able to show that such an approach gives better approximation to the original signal $\mathbf{x}_*$.

# 1 Background: A Constrained Minimal $\ell_1$-norm Problem

The ingenious Nyquist-Shannon Sampling Theorem addresses the question about the possibility to recover any signal using finite number of sampling (measurements); however, according to the theorem, the number of sampling to take is two times the highest frequency in a signal. The pursuit of improvement on reducing the sampling rates have been non-stop since 1949; and significant progress has been made especially regarding the sparse signal recovery. In fact, hundreds of papers in the Compressed Sensing community have been published to reduce the sampling rates to a significant level. The theory regarding sparse signal recovery has been proven and perfected in a series of papers starting in mid 2004. See [4, 9, 13, 10, 7, 6, 5] for complete details.

Let us consider a target signal $\mathbf{x}_* \in \mathbf{R}^n$, with sparsity $l$, which simply means $l$ non-zero entries. let $m$ represent the number of linear non-adaptive[1] measurements one wants to take:

*mzhong1@umd.edu
†tadmor@cscamm.umd.edu

[1]Means of measurements does not depend on $\mathbf{x}$.

let $\mathbf{a}_i \in \mathbf{R}^n$ be a basis vector. Measurements are made by taking the usual Euclidean inner product between $\mathbf{a}_i$ and $\mathbf{x}$, that is $\langle \mathbf{a}_i, \mathbf{x} \rangle_{\ell_2}$. Let $A$ be a concatenation of the $m$ (randomly picked) basis vectors $\mathbf{a}_i \in \mathbf{R}^n$, for $i = 1, \ldots, m$; then $A = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \cdots \\ \mathbf{a}_m^T \end{bmatrix} \in \mathbf{R}^{m \times n}$. Two possible scenarios exit for choosing the matrix $A$: it is either prescribed by a collection of basis vectors from a specific transformation or constructed with certain properties, namely the Restricted Isometry Properties (or its equivalence). The problem which we mentioned is basically asking that with the knowledge of only $A\mathbf{x}_*$ and $m \ll n$, can $\mathbf{x}_*$ be recovered? And if so, how? The answer in general is no. However when $l$ is small, equivalently saying that $\mathbf{x}_*$ is sparse, then the possibility of full recovery is high according to the compressive sensing principle. Moreover the sparse signal $\mathbf{x}_*$ can be recovered through the following constrained minimal $\ell_p$-norm problem:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \{ ||\mathbf{x}||_{\ell_p} \mid A\mathbf{x} = A\mathbf{x}_* \} \tag{1}$$

The question remains: what would be an appropriate $p$ for the purpose of recovering a sparse signal? As stated in [13], the sparsity of a vector $\mathbf{x}$ is usually defined as (for some $0 < p < 2$):

$$||\mathbf{x}||_{\ell_p} \equiv (\sum_i |x_i|^p)^{\frac{1}{p}} \leq l \tag{2}$$

(We will also consider the border line cases $p = 0$ and $p = 2$). According to [11, 12], $\ell_p$-norm with $0 \leq p < 1$ are intuitive ways to preserve sparsity in the mathematical setting. Let us consider an extreme case when $p = 0$, when the $\ell_0$-norm simply measures the number of non-zero elements in a vector $\mathbf{x}$, directly imposing sparsity. Letting $p = 0$ in (1), we obtain:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \{ ||\mathbf{x}||_{\ell_0} \mid A\mathbf{x} = A\mathbf{x}_* \} \tag{3}$$

The minimizer $\mathbf{x}_{\ell_0}$ will the same as $\mathbf{x}_*$ if we take $m = l + 1$ measurements[2]. However, it is shown in [20] that (3) is NP hard and requires techniques from combinatorial optimization. Hence it is numerically appropriate to consider a convex (1) by using either the $\ell_1$ or $\ell_2$-norm[3], since (1) is considered a convex optimization problem for $p \geq 1$. With a convex (1), we are able to use linear programming to find the minimizer. From now on, we will denote the measurement results $A\mathbf{x}_*$ as $\mathbf{b}$ and turn to the following Constrained Least Square Problem:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \{ ||\mathbf{x}||_{\ell_2} \mid A\mathbf{x} = \mathbf{b} \} \tag{4}$$

It is also called a constrained minimum norm problem. An analytic solution exists and is $\mathbf{x}_{\ell_2} = A^T(AA^T)^{-1}\mathbf{b}$, since $A$ is a fat matrix, $AA^T$ has an inverse when $A$ has full rank[4]. Apparently $A\mathbf{x}_{\ell_2} = AA^T(AA^T)^{-1}\mathbf{b} = \mathbf{b}$. Let $\mathbf{x}$ be another solution of the linear system, that is $A\mathbf{x} = \mathbf{b}$. Then we claim that $\mathbf{x} - \mathbf{x}_{\ell_2}$ is perpendicular to $\mathbf{x}_{\ell_2}$. To Justify our claim, observe the following:

$$\begin{aligned} \langle \mathbf{x} - \mathbf{x}_{\ell_2}, \mathbf{x}_{\ell_2} \rangle_{\ell_2} &= \langle \mathbf{x} - \mathbf{x}_{\ell_2}, A^T(AA^T)^{-1}\mathbf{b} \rangle_{\ell_2} = \langle A(\mathbf{x} - \mathbf{x}_{\ell_2}), (AA^T)^{-1}\mathbf{b} \rangle_{\ell_2} \\ &= \langle \mathbf{0}, (AA^T)^{-1}\mathbf{b} \rangle_{\ell_2} = 0 \end{aligned}$$

[2] Bresler; Wakin et al

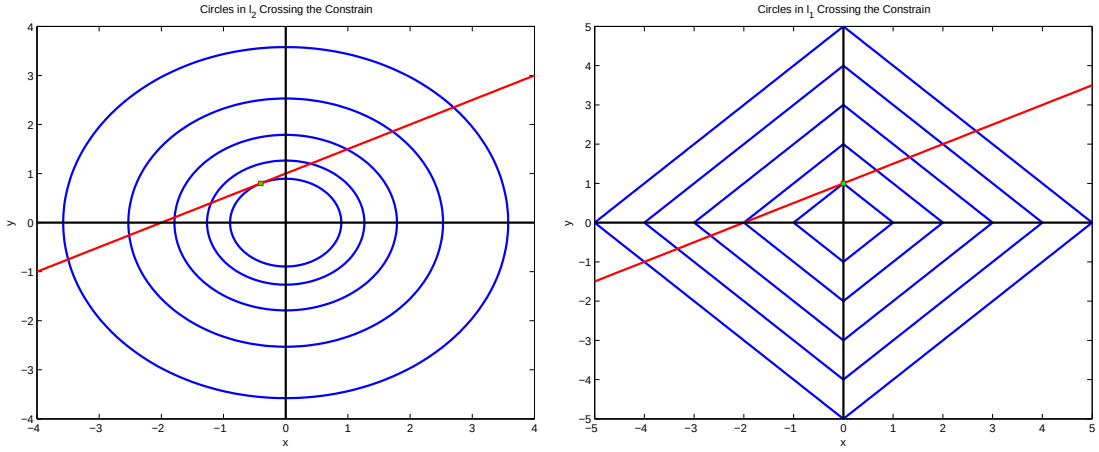
[3] We do not consider $0 < p < 1$, since such $\ell_p$-norm is not convex.

[4] Otherwise $(AA^T)^{-1}$ is computed using SVD.

Hence by the Pythagorean Theorem in $\mathbf{R}^n$:

$$||\mathbf{x}||_{\ell_2}^2 = ||\mathbf{x} - \mathbf{x}_{\ell_2} + \mathbf{x}_{\ell_2}||_{\ell_2}^2 = ||\mathbf{x} - \mathbf{x}_{\ell_2}||_{\ell_2}^2 + ||\mathbf{x}_{\ell_2}||_{\ell_2}^2 \geq ||\mathbf{x}_{\ell_2}||_{\ell_2}^2$$

Therefore $||\mathbf{x}||_{\ell_2} \geq ||\mathbf{x}_{\ell_2}||_{\ell_2}$, with equality realized only when $\mathbf{x} = \mathbf{x}_{\ell_2}$. $\mathbf{x}_{\ell_2}$ is indeed the optimizer. Unfortunately, in most cases, $\mathbf{x}_{\ell_2}$ is not going to be sparse, despite the fact that $\mathbf{x}_{\ell_2}$ is relatively easy to compute. Let us explain why $\mathbf{x}_{\ell_2}$ is dense in the two dimensional setting: the solution set of $A\mathbf{x} = \mathbf{b}$ would consist of a straight line, and the level sets $\mathcal{B}(r) = \{\mathbf{x} \in \mathbf{R}^n \mid ||\mathbf{x}||_{\ell_2} \leq r\}$ are circles of radius $r$ centered at the origin. The constrained least square problem is asking that out of all the possible points on the said line, which one would be on the level set $\mathcal{B}(r)$ with the smallest $r$. As it is shown below, most of the time, the intersection happens in the interior of a quadrant, thus making $\mathbf{x}_{\ell_2}$ not sparse. However if the $\ell_1$-norm is used, since the level sets $\mathcal{D}(r) = \{\mathbf{x} \in \mathbf{R}^n \mid ||\mathbf{x}||_{\ell_1} \leq r\}$ are diamonds, The intersection can happen at one of the axis, thus making the minimizer sparse.




(a) $\ell_2$ Minimizer (b) $\ell_1$ Minimizer

Figure 1: Comparison between the two minimizers

With $\mathbf{x}_{\ell_2}$ not satisfying the sparsity requirement, we would like to consider the $\ell_1$-norm instead. In fact, the $\ell_1$ technique is widely used to induce sparsity on solutions. Hence we have:

$$\min_{\mathbf{x} \in \mathbf{R}^n} \{||\mathbf{x}||_{\ell_1} \mid A\mathbf{x} = \mathbf{b}\} \tag{5}$$

Under certain conditions, the solution $\mathbf{x}_{\ell_1}$ from (5) recovers the original signal $\mathbf{x}_*$ and actually coincides with $\mathbf{x}_{\ell_0}$ from (3). The authors in [4] proved such equivalence with the matrix $A$ being a Fourier Transform matrix. They later generalized the proof to any matrix $A$ with the Restricted Isometry Property in [7, 6, 5]. Then separately the author in [13, 10] showed that when $A$ has the CS1-CS3 properties, $\mathbf{x}_*$ can be recovered from solving (5). Moreover, all of the papers require that the recovery can be achieved only when $m \simeq l * log(n) \ll n$.

**Remark 1.1.** *However the $\ell_1$ problem is not well-posed. Let us begin our argument by considering a family of more general problems:*

$$\min_{\mathbf{x} \in \mathbf{R}^n} \{J(\mathbf{x}) \mid A\mathbf{x} = \mathbf{b}\} \tag{6}$$

*Where $J(\cdot)$ is a continuous and convex energy functional defined on $\mathbf{R}^n$. Let $J(\cdot)$ also be coercive, that is whenever $||\mathbf{x}||_{\ell_2} \to \infty$, $J(\mathbf{x}) \to \infty$. By convex analysis, the solution set of* (6) *is nonempty and convex. And if $J(\cdot)$ is strictly or strongly convex, the solution set contains only one point (unique solution, such as $\mathbf{x}_{\ell_2}$ from* (4)*). However $J(\mathbf{x}) = ||\mathbf{x}||_{\ell_1}$ is not strictly convex, the solution of* (5) *might not be unique.*

## 1.1 Tikhonov Regularization

With (5) possibly being ill-posed, we will add Tikhonov Regularization to (5) and bring the constraint into the minimization as a regularization. Such regularization, which was proposed and developed in [25, 26], is specially designed to tackle ill-posed problems. We obtain the following:

$$\min_{\mathbf{x}\in\mathbf{R}^n}\{||\mathbf{x}||_{\ell_1}+\frac{\lambda}{2}||\mathbf{b}-A\mathbf{x}||_{\ell_2}^2\} \tag{7}$$

The regularization parameter dictates the distance between the solution of (7), call it $\mathbf{x}_{\ell_1}(\lambda)$, and $\mathbf{x}_{\ell_1}$. See [17] for a complete proof. When $\lambda$ is small, we are basically minimizing $||\mathbf{x}||_{\ell_1}$, which might lead to $\mathbf{x}_{\ell_1}(\lambda)=\mathbf{0}$; when $\lambda$ is large, we are putting more emphasis on the $\ell_2$ part, thus forcing $\mathbf{x}_{\ell_1}(\lambda)$ to get closer to the hyper plane $\mathcal{H}=\{\mathbf{x}\in\mathbf{R}^n \mid A\mathbf{x}=\mathbf{b}\}$, hence getting closer to $\mathbf{x}_{\ell_1}$. Without any *a priori* information of $\mathbf{x}_{\ell_1}$, we want to know certain properties of the solution $\mathbf{x}_{\ell_1}(\lambda)$. Let $\mathbf{r}(\mathbf{x}_{\ell_1}(\lambda))=\mathbf{b}-A\mathbf{x}_{\ell_1}(\lambda)$, and we apply the theorems in [23, 24] with $U=\ell_1, W=\ell_2, q=2$ and $T=A$. Thus we are able to show that $\mathbf{x}_{\ell_1}(\lambda)$ and $\mathbf{r}(\mathbf{x}_{\ell_1}(\lambda))$ satisfy the following:

**Theorem 1.2** (Validation Principles)**.**

$$\langle\mathbf{x}_{\ell_1}(\lambda),A^T\mathbf{r}(\mathbf{x}_{\ell_1}(\lambda))\rangle_{\ell_2} = ||\mathbf{x}_{\ell_1}(\lambda)||_{\ell_1}\,||A^T\mathbf{r}(\mathbf{x}_{\ell_1}(\lambda))||_{\infty} \tag{8}$$

$$||A^T\mathbf{r}(\mathbf{x}_{\ell_1}(\lambda))||_{\ell_\infty} = \frac{1}{\lambda} \tag{9}$$

(8) and (9) will be used to verify the implementation of various $\ell_1$ solvers later on.

**Remark 1.3.** *Normally by Hölder's inequality,* $\langle\mathbf{x},A^T\mathbf{r}(\mathbf{x})\rangle_{\ell_2}\le||\mathbf{x}||_1\,||A^T\mathbf{r}(\mathbf{x})||_\infty$. *Equality is only realized when* $\mathbf{x}=\mathbf{x}_{\ell_1}(\lambda)$. *Therefore we call this pair* $(\mathbf{x}_{\ell_1}(\lambda),\,\mathbf{r}(\mathbf{x}_{\ell_1}(\lambda)))$ *an extremal pair. Meanwhile, since* $\mathbf{r}(\mathbf{x}_{\ell_1})=\mathbf{0}$ *together with* (9)*,* $\mathbf{x}_{\ell_1}(\lambda)=\mathbf{x}_{\ell_1}$ *happens when* $\lambda\to\infty$ *(*$A^T$ *has full row rank).*

Using straightforward calculus of variation technique, one can derive a non-linear equation from (7), which is:

$$\mathbf{sgn}(\mathbf{x})+\lambda A^T\mathbf{r}(\mathbf{x})=\mathbf{0} \tag{10}$$

Where the $\text{sgn}(a)=\begin{cases}1, & a>0\\ -1, & a<0\end{cases}$; and $\mathbf{sgn}(\cdot)$ is defined component wise for vectors. However, $\text{sgn}(a)$ is not defined at $a=0$. Such delicacy makes (10) hard to solve directly; furthermore, the solution set of (10) is contained in the solution set of (7) due to non-differentiability property of $\text{sgn}(\cdot)$ at 0.

# 2 Hierarchical Reconstruction

Following a similar arguments by the authors in [23], we present the following motivation for the Hierarchical Reconstruction algorithm.

## 2.1 Motivation

Let $\mathbf{x}_\lambda$ and $\mathbf{r}(\mathbf{x}_\lambda)$ be an extremal pair such that they solve (7), that is:

$$[\mathbf{x}_\lambda,\mathbf{r}(\mathbf{x}_\lambda)]=\underset{A\mathbf{x}+\mathbf{r}(\mathbf{x})=\mathbf{b}}{\arg\min}\{||\mathbf{x}||_{\ell_1}+\frac{\lambda}{2}||\mathbf{r}(\mathbf{x})||_{\ell_2}^2\} \tag{11}$$

The pair $[\mathbf{x}_\lambda, \mathbf{r}(\mathbf{x}_\lambda)]$ also decompose $\mathbf{b}$ into two parts: the recovered sparse signal $\mathbf{x}_\lambda$ and residual $\mathbf{r}(\mathbf{x}_\lambda)$ under the given scale $\lambda$. However, the difference between these two components is scale dependent - whatever is considered as residual at a given scale $\lambda$ contains significant information when viewed under a refined scale, say $2\lambda$:

$$[\mathbf{x}_{2\lambda}, \mathbf{r}(\mathbf{x}_{2\lambda})] = \underset{A\mathbf{x}+\mathbf{r}(\mathbf{x})=\mathbf{r}(\mathbf{x}_\lambda)}{\arg\min} \{||\mathbf{x}||_{\ell_1} + \frac{2\lambda}{2}||\mathbf{r}(\mathbf{x})||_{\ell_2}^2\} \tag{12}$$

Since $\mathbf{b} = A(\mathbf{x}_\lambda + \mathbf{x}_{2\lambda}) + \mathbf{r}(\mathbf{x}_{2\lambda}) \approx A(\mathbf{x}_\lambda + \mathbf{x}_{2\lambda})$, we obtain a better two-scale approximation to $\mathbf{b}$; noise below scale $\frac{1}{2\lambda}$ remains unresolved in $\mathbf{r}(\mathbf{x}_{2\lambda})$. This process of (11) and (12) can continue. Beginning with an initial scale $\lambda = \lambda_0$,

$$[\mathbf{x}_0, \mathbf{r}(\mathbf{x}_0)] = \underset{A\mathbf{x}+\mathbf{r}(\mathbf{x})=\mathbf{b}}{\arg\min} \{||\mathbf{x}||_{\ell_1} + \frac{\lambda_0}{2}||\mathbf{r}(\mathbf{x})||_{\ell_2}^2\}$$

we continue in this iterative manner for the decomposition of the dyadic refinement step (12),

$$[\mathbf{x}_{j+1}, \mathbf{r}(\mathbf{x}_{j+1})] = \underset{A\mathbf{x}+\mathbf{r}(\mathbf{x})=\mathbf{r}(\mathbf{x}_j)}{\arg\min} \{||\mathbf{x}||_{\ell_1} + \frac{\lambda_j}{2}||\mathbf{r}(\mathbf{x})||_{\ell_2}^2\}, \quad j = 0, 1, 2, \ldots \tag{13}$$

generating, after $k$ such steps, the following **Hierarchical Decomposition** of $\mathbf{b}$:

$$\begin{aligned} \mathbf{b} &= A\mathbf{x}_0 + \mathbf{r}(\mathbf{x}_0) \\ &= A\mathbf{x}_0 + A\mathbf{x}_1 + \mathbf{r}(\mathbf{x}_1) \\ &= \ldots\ldots \\ &= A\mathbf{x}_0 + A\mathbf{x}_1 + \ldots + A\mathbf{x}_k + \mathbf{r}(\mathbf{x}_k). \end{aligned} \tag{14}$$

We arrive at a new multi-scale signal decomposition, $\mathbf{b} \approx A\mathbf{x}_0 + A\mathbf{x}_1 + \ldots + A\mathbf{x}_k$, with a residual $\mathbf{r}(\mathbf{x}_k)$. As $k$ increases, we reconstruct the signal $\mathbf{x} = \sum_{j=0}^{k} \mathbf{x}_j$ with increasing accuracy. What remains a question is whether the sparsity of $\mathbf{x}$ stays relatively unchaged.

## 2.2 The Algorithm

We present the following algorithm:

**Algorithm 1** Hierarchical Reconstruction of Sparse Signals

**Require:** $A$ and $\mathbf{b}$, pick $\lambda_0$
  Initialize: $\mathbf{x}_0 = 0$, $\mathbf{r}(\mathbf{x}_0) = \mathbf{b} - A\mathbf{x}_0$, and $j = 0$
  **while** $j \leq J$ **do**
    $\mathbf{x}_{j+1} \coloneqq \underset{\mathbf{x}\in\mathbf{R}^n}{\arg\min} \{||\mathbf{x}||_{\ell_1} + \frac{\lambda_j}{2}||\mathbf{r}(\mathbf{x}_j)||_{\ell_2}^2\}$
    $\mathbf{r}(\mathbf{x}_{j+1}) = \mathbf{r}(\mathbf{x}_j) - A\mathbf{x}_{j+1}$
    $\lambda_{j+1} = 2\lambda_j$;
    $j = j + 1$;
  **end while**
**Ensure:** $\mathbf{x} = \sum_{j=0}^{J} \mathbf{x}_j$;

According to [23], if $||A^T\mathbf{b}||_{\ell_\infty} \leq \frac{1}{\lambda}$, then the optimizer of (7) is $\mathbf{0}$. To avoid the $\mathbf{0}$ optimizer, we will pick an initial $\lambda_0$ such that

$$\frac{1}{||A^T\mathbf{b}||_{\ell_\infty}} \leq \lambda_0 \leq \frac{2}{||A^T\mathbf{b}||_{\ell_\infty}} \tag{15}$$

We also pick a stopping $\lambda_J$ suggested in [21]. With a suitable solver for (7) the algorithm 1 is very intuitive to implement.

# 3 Implementation

We choose the Gradient Projection for Sparse Reconstruction method and A Fixed-Point Continuation method as built-in solvers for (7). We have singled these two methods out for their robustness and efficiency over IST [8, 15], l1_ls package [19], $\ell_1$-magic toolbox[3], and homotopy method [14].

## 3.1 Gradient Projection for Sparse Reconstruction

The **Gradient Projection for Sparse Reconstruction** algorithm is proposed in [16] solve the following:

$$\min_{\mathbf{x}\in\mathbf{R}^n}\{\tau||\mathbf{x}||_{\ell_1}+\frac{1}{2}||\mathbf{b}-A\mathbf{x}||_{\ell_2}^2\} \tag{16}$$

Compared to (7), (16) puts the regularization on the $\ell_1$ term instead. Since $\tau>0$ and $\lambda>0$, we have $\tau=\frac{1}{\lambda}$. Let $(a)_+=\begin{cases} a, & a\geq 0 \\ 0, & \text{otherwise}\end{cases}$, and then $\mathbf{u}=(\mathbf{x})_+$, $\mathbf{v}=(-\mathbf{x})_+$. With $\mathbf{u}$ and $\mathbf{v}$ substituted back into (16), we can transform (16) into a linear problem by first letting:

$$\mathbf{z}=\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix},\quad \mathbf{y}=A^T\mathbf{b},\quad \mathbf{c}=\tau\mathbf{1}_{2n}+\begin{bmatrix}-\mathbf{y}\\ \mathbf{y}\end{bmatrix},\quad B=\begin{bmatrix}A^TA & -A^TA\\ -A^TA & A^TA\end{bmatrix}$$

then we obtain,

$$\min_{\mathbf{z}\in\mathbf{R}^{2n}}\{F(\mathbf{z})\equiv\mathbf{c}^T\mathbf{z}+\frac{1}{2}\mathbf{z}^TB\mathbf{z}\;\big|\;\mathbf{z}\geq 0\} \tag{17}$$

We will present two algorithms, which use gradient projection techniques to find the minimizer of (16). They both pick the descent direction $\mathbf{d}^{(k)}$ as (with $\alpha^{(k)}>0$):

$$\mathbf{d}^{(k)}=(\mathbf{z}^{(k)}-\alpha^{(k)}\nabla F(\mathbf{z}^{(k)}))_+-\mathbf{z}^{(k)} \tag{18}$$

And they both update the next iterate $\mathbf{z}^{(k+1)}$ as (with $\nu^{(k)}$):

$$\mathbf{z}^{(k+1)}=\mathbf{z}^{(k)}+\nu^{(k)}\mathbf{d}^{(k)} \tag{19}$$

The two approaches differ by choosing different $\alpha^{(k)}$ and $\lambda^{(k)}$. We will start with the basic approach:

**Algorithm 2** GPSR Basic

**Require:** $A$, $\mathbf{b}$, $\tau$, and $\mathbf{z}^{(0)}$, pick $\beta\in(0,1)$ and $\mu\in(0,1/2)$
  Initialize: $k=0$;
  **while** A stopping criteria is not satisfied **do**
    Compute $\alpha_0=\arg\min_{\alpha\in\mathbf{R}^1}\{F(\mathbf{z}^{(k)}-\alpha\mathbf{G}^{(k)})\}$
    Let $\alpha^{(k)}$ be the first in the sequence: $\alpha_0$, $\beta\alpha_0$, $\beta^2\alpha_0$, ..., such that
  $F((\mathbf{z}^{(k)}-\alpha^{(k)}\nabla F(\mathbf{z}^{(k)}))_+)\leq F(\mathbf{z}^{(k)})-\mu\nabla F(\mathbf{z}^{(k)})^T(\mathbf{z}^{(k)}-(\mathbf{z}^{(k)}-\alpha^{(k)}\nabla F(\mathbf{z}^{(k)}))_+)$
    Set $\mathbf{z}^{(k+1)}=(\mathbf{z}^{(k)}-\alpha^{(k)}\nabla F(\mathbf{z}^{(k)}))_+$
    Update $k=k+1$
  **end while**
**Ensure:** $\mathbf{z}^{(K)}(\tau):=\arg\min_{\mathbf{z}\in\mathbf{R}^{2n}}\{F(\mathbf{z})\equiv\mathbf{c}^*\mathbf{z}+\frac{1}{2}\mathbf{z}^*B\mathbf{z}\;\big|\;\mathbf{z}\geq 0\}$;

The basic approach would pick $\alpha^{(k)}$ from a variable line search such that $F((\mathbf{z}^{(k)}-\alpha^{(k)}\nabla F(\mathbf{z}^{(k)}))_+)$ is minimal and pick $\nu^{(k)}=1$. We will start with an initial guess $\mathbf{z}^{(0)}=\frac{1}{2n}\mathbf{1}_{2n}$, that is $||\mathbf{z}^{(0)}||_{\ell_1}=1$. Since $F(\cdot)$ is quadratic, there is formula for calculating $\alpha_0$, namely $\alpha_0=\frac{(\mathbf{G}^{(k)})^T\nabla F(\mathbf{z}^{(k)})}{(\mathbf{G}^{(k)})^T B\mathbf{G}^{(k)}}$ and restricted to $[\alpha_{\min},\alpha_{\max}]$. $\beta$ is a back-tracking parameter to pick the optimal step size for gradient descent. $\mu$ is used to make sure that $F(\cdot)$ is decreased sufficiently from the "Armijo rule along the projection arc" by [2, p. 226] and $\mathbf{G}^{(k)}$ is a projected gradient, defined component wise:

$$G_i^{(k)}=\begin{cases}(\nabla F(\mathbf{z}^{(k)}))_i, & \text{if} \quad \mathbf{z}_i^{(k)}>0 \quad \text{or} \quad (\nabla F(\mathbf{z}^{(k)}))_i<0\\ 0, & \text{otherwise}\end{cases} \tag{20}$$

Thus $(\mathbf{G}^{(k)})^T\nabla F(\mathbf{z}^{(k)})=(\mathbf{G}^{(k)})^T\mathbf{G}^{(k)}$. The second approach is based on Barzilai and Broweign's paper [1] and BCQP approach [22] to avoid the decaying convergence rate property of some steepest descent methods. Each time an update is calculated as $\mathbf{d}^{(k)}=-H_k^{-1}\nabla F(\mathbf{z}^{(k)})$, where $H_k$ is the Hessian of $F(\mathbf{z}^{(k)})$. $H_k$ is approximated by a multiple of the identity $H_k\approx\rho^{(k)}I$, where $\rho^{(k)}$ is chosen as:

$$\nabla F(\mathbf{z}^{(k)})-\nabla F(\mathbf{z}^{(k-1)})\approx\rho^{(k)}\big(\mathbf{z}^{(k)}-\mathbf{z}^{(k-1)}\big)$$

Then set $\alpha^{(k)}=(\rho^{(k)})^{-1}$ and restricted to the interval $[\alpha_{\min},\alpha_{\max}]$; and let $\nu^{(k)}\in[0,1]$ be the exact minimizer of $F(\mathbf{z}^{(k)}+\nu\mathbf{d}^{(k)})$, then we have to following algorithm:

**Algorithm 3** GPSR Barzilai Browein

**Require:** $A$, $\mathbf{b}$, $\tau$, $\mathbf{z}^{(0)}$, $\alpha_{\min}$, $\alpha_{\max}$, and pick $\alpha^{(0)}\in[\alpha_{\min},\alpha_{\max}]$
  Initialize: $k=0$;
  **while** A stopping criteria is not satisfied **do**
    Compute $\mathbf{d}^{(k)}=(\mathbf{z}^{(k)}-\alpha^{(k)}\nabla F(\mathbf{z}^{(k)}))_+-\mathbf{z}^{(k)}$
    Compute $\omega^{(k)}=(\mathbf{d}^{(k)})^T B\mathbf{d}^{(k)}$
    **if** $\omega^{(k)}=0$ **then**
      Set $\alpha^{(k+1)}=\alpha_{\max}$ and $\nu^{(k)}=1$
    **else**
      Compute $\alpha^{(k)}=\text{mid}(\alpha_{\min},\frac{||\mathbf{d}^{(k)}||_{\ell_2}^2}{\omega^{(k)}},\alpha_{\max})$ and $\nu^{(k)}=\text{mid}(0,\frac{-(\mathbf{d}^{(k)})^T\nabla F(\mathbf{z}^{(k)})}{(\mathbf{d}^{(k)})^T B\mathbf{d}^{(k)}},1)$
    **end if**
    set $\mathbf{z}^{(k+1)}=\mathbf{z}^{(k)}+\nu^{(k)}\mathbf{d}^{(k)}$
    Update $k=k+1$
  **end while**
**Ensure:** $\mathbf{z}^{(K)}:=\underset{\mathbf{z}\in\mathbf{R}^{2n}}{\arg\min}\{F(\mathbf{z})\equiv\mathbf{c}^T\mathbf{z}+\frac{1}{2}\mathbf{z}^T B\mathbf{z}\;\big|\;\mathbf{z}\geq 0\}$;

We will pick the same initial guess $\mathbf{z}^{(0)}$ as it is used in algorithm 2. As it is stated in [16], we will consider these situations for choosing a suitable stopping criteria:

- The approximation $\mathbf{z}$ is close to a solution $\mathbf{z}_*$.
- The function value $F(\mathbf{z})$ is close to $F(\mathbf{z}_*)$.
- The non-zero components of the approximation $\mathbf{z}$ are close to the non-zero components of $\mathbf{z}_*$.

Suggested in [16], we will run the codes with the following stopping criterion:

- $||\mathbf{z} - (\mathbf{z} - \bar{\alpha}\nabla F(\mathbf{z}))_+||_2 \leq$ tolP, where $\bar{\alpha}$ is a positive constant, and tolP is a small parameter. Note that when $\mathbf{z}$ is optimal, the left hand side is zero.

- $||\mathbf{min}(\mathbf{z}, \nabla F(\mathbf{z}))||_2 \leq$ tolP, the $\mathbf{min}(\cdot)$ is taken component wise. It is motivated by perturbation results from linear complementarity problems (LCP). LCP results show that there is a constant $C_{LCP}$ such that $\text{dist}(\mathbf{z}, \mathcal{S}) \leq C_{CLP}||\mathbf{min}(\mathbf{z}, \nabla F(\mathbf{z}))||_2$, where $\mathcal{S}$ is a solution set of (17). When $||\mathbf{min}(\mathbf{z}, \nabla F(\mathbf{z}))||_2 \leq$ tolP, we are forcing $\mathbf{z}$ to be within a certain distance of the solution set $\mathcal{S}$.

- $\left|\frac{1}{2}||\mathbf{b} - A\mathbf{x}||_{\ell_2}^2 + \tau||\mathbf{x}||_{\ell_1} + \frac{1}{2}\mathbf{s}^T\mathbf{s} + \mathbf{b}^T\mathbf{s}\right| \leq$ tolP, where $\mathbf{s}$ is a minimizer of the dual problem of (16),

$$\max_{\mathbf{s}}\{-\frac{1}{2}\mathbf{s}^T\mathbf{s} - \mathbf{b}^T\mathbf{s} \mid -\tau\mathbf{1}_n \leq A^T\mathbf{s} \leq \tau\mathbf{1}_n\} \tag{21}$$

  Therefore on each iterate we resemble $\mathbf{x} = \mathbf{u} - \mathbf{v}$ and compute $\mathbf{s} = \tau\frac{A\mathbf{x}-\mathbf{b}}{||A^T(A\mathbf{x}-\mathbf{b})||_{\ell_\infty}}$, and substitute the pair back into the left hand side of the inequality and to check to see if the left hand side is less than tolP.

- $|\mathcal{C}_k|/|\mathcal{I}_k| \leq$ tolP, where $\mathcal{I}_k = \{i \mid \mathbf{z}_i^{(k)} \neq 0\}$ and $\mathcal{C}_k = \{i \mid i \in \mathcal{I}_k \oplus \mathcal{I}_{k-1}\}$[5]. Such ratio takes into account that changes in the non-zero entries in the approximation $\mathbf{z}$ are going to be small when $\mathbf{z}$ is close to $\mathbf{z}_*$.

When the computed solution $\mathbf{z} = \begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}$ is converted back to $\mathbf{x}_{GPSR} = \mathbf{u} - \mathbf{v}$, we might perform a debiasing step. We will keep the zero components of $\mathbf{x}_{GPSR}$ fixed and minimize the regularization part $||\mathbf{b} - A\mathbf{x}||_2^2$ using a Conjugate Gradient algorithm (with a restriction) and terminates when $||\mathbf{b} - A\mathbf{x}||_2^2 \leq \text{tolD}||\mathbf{b} - A\mathbf{x}_{GPSR}||_2^2$ with a small tolD and a preset maxIterD (maximum iterations allowed).

**Remark 3.1.** *Although GPSR algorithms have increased the system size from $n$ to $2n$, the matrix-vector multiplication can still be done at the $O(n)$. We can simply the matrix-vector multiplication by:*

$$B\mathbf{z} = \begin{bmatrix} A^TA(\mathbf{u} - \mathbf{v}) \\ -A^TA(\mathbf{u} - \mathbf{v})\end{bmatrix}, \quad \mathbf{c}^T\mathbf{z} = \tau\mathbf{1}_n^T(\mathbf{u} + \mathbf{v}) - \mathbf{y}^T(\mathbf{u} - \mathbf{v}), \quad \mathbf{z}^TB\mathbf{z} = ||A(\mathbf{u} - \mathbf{v})||_2$$

*Hence we have:*

$$F(\mathbf{z}) = \tau\mathbf{1}_n^T(\mathbf{u} + \mathbf{v}) - \mathbf{y}^T(\mathbf{u} - \mathbf{v}) + \frac{1}{2}||A(\mathbf{u} - \mathbf{v})||_2, \quad \nabla F(\mathbf{z}) = \begin{bmatrix}\tau\mathbf{1}_n - \mathbf{y} + A^TA(\mathbf{u} - \mathbf{v})\\ \tau\mathbf{1}_n + \mathbf{y} - A^TA(\mathbf{u} - \mathbf{v})\end{bmatrix}$$

*$\mathbf{1}_n^T(\mathbf{u} + \mathbf{v})$ can be done using the MATLAB$^{TM}$ built-in summation function for vectors. We do not compute the matrix-matrix product $A^TA$ directly; instead we will do $A(\mathbf{u} - \mathbf{v})$ first then $A^T(\cdot)$.*

Theorem 1 in [16] states that the sequence of $\{\mathbf{z}^{(k)}\}$ generated by the GPSR algorithms either terminate at a solution of (17) or converge to a solution of (17) at an R-linear rate.

[5] $\oplus$ stands for the exclusive union of two sets

## 3.2 Fixed-Point Continuation Method

In [18], the **Fixed-Point Continuation** algorithm is developed to solve the following:

$$\min_{\mathbf{x}\in\mathbf{R}^n}\{||\mathbf{x}||_1+\frac{\lambda}{2}||\mathbf{b}-A\mathbf{x}||_D^2\} \tag{22}$$

Where $||\mathbf{x}||_D \coloneqq \sqrt{\mathbf{x}^T D\mathbf{x}}$ and $D$ is a Symmetric Positive Definite matrix. In this project, we take $D=I$. Let $f:\mathbf{R}^n\to\mathbf{R}$ and $\mathbf{g}:\mathbf{R}^n\to\mathbf{R}^n$ be defined as followed:

$$\begin{aligned} f(\mathbf{x}) &\coloneqq \frac{1}{2}||\mathbf{b}-A\mathbf{x}||_{\ell_2}^2 \\ \mathbf{g}(\mathbf{x}) &\coloneqq \nabla f(\mathbf{x}) = A^T(A\mathbf{x}-\mathbf{b}) \end{aligned}$$

And consider two mappings $\mathbf{s}_\xi$ and $\mathbf{h}$, both from $\mathbf{R}^n$ to $\mathbf{R}^n$, defined as (for any $\eta>0$):

$$\mathbf{h}(\mathbf{x}) \coloneqq \mathbf{x}-\eta\mathbf{g}(\mathbf{x}) \tag{23}$$

$$\mathbf{s}_\xi(\mathbf{x}) \coloneqq \mathbf{sgn}(\mathbf{x})\odot\max\{|\mathbf{x}|-\xi,0\} \tag{24}$$

where $\xi=\frac{\eta}{\lambda}$ and $\odot$ is a component wise multiplication for vectors. And then consider the following fixed point equation:

$$\mathbf{x}^{(k+1)}=\mathbf{s}_\xi\circ\mathbf{h}(\mathbf{x}^{(k)}) \tag{25}$$

**Remark 3.2.** *As it is proven in [18], if* $\mathbf{x}_{\ell_1}(\lambda)$ *solves* (7)*, then* $\mathbf{0}\in\boldsymbol{sgn}(\mathbf{x}_{\ell_1}(\lambda))+\lambda\mathbf{g}(\mathbf{x}_{\ell_1}(\lambda))$ *and vice versa. And if* $\mathbf{x}_{sgn}(\lambda)$ *is a fixed point of* (25)*, then* $\mathbf{x}_{sgn}(\lambda)\in\boldsymbol{sgn}(\mathbf{x}_{sgn}(\lambda))+\lambda\mathbf{g}(\mathbf{x}_{sgn}(\lambda))$ *and vice versa. Therefore* $\mathbf{x}_{\ell_1}(\lambda)=\mathbf{x}_{sgn}(\lambda)$.

Let $\varrho_{\max}$ be the maximum eigenvalue of the Hessian of $f(\mathbf{x})$, namely $H(\mathbf{x})=A^TA$. As it is shown in [18], we have to maintain $\eta\in(0,\frac{2}{\varrho_{\max}})$[6] in order to have convergence results. Under this setting, [18] proposes the following algorithm:

**Algorithm 4** Fixed Point Continuation Method

**Require:** $A$, $\mathbf{b}$, $\lambda$, pick $\mathbf{x}^{(0)}$, set $\bar{\mu}=\lambda$
  Select: $0<\mu_1<\mu_2<\cdots<\mu_L=\bar{u}$
  **for** $\mu=\mu_1,\mu_2,\cdots,\mu_L$ **do**
    **while** A convergence test is not satisfied **do**
      Select $\eta$ and set $\xi=\frac{\eta}{\mu}$
      $\mathbf{x}^{(k+1)}=\mathbf{s}_\xi\circ\mathbf{h}(\mathbf{x}^{(k)})$
    **end while**
  **end for**
**Ensure:** $\mathbf{x}_{\text{sgn}}(\lambda)$

Although the fixed point iteration is simple, the algorithm depends on a suitable choice of $\eta$, an appropriate sequence of $\mu_i$'s and an initial guess $\mathbf{x}^{(0)}$, and even the stopping criterion are problem dependent. All of these requirement makes the algorithm harder to apply to general problems.

[6] To have faster convergence, we have to have $\eta\in[\frac{1}{\varrho_{max}},\frac{2}{\varrho_{max}})$

### 3.3 Implementation Platform and Memory Allocation

Codes will be written in MATLAB™ for GPSR Basic, GPSR Barzilai Borwein, FPC, and the whole HRoSS algorithm. When time permits, parallel codes will be written in C. The version of the MATLAB™ running on my personal computer is: $7.12.0.635(R2011a)$. It is installed on a copy of the Windows7™ Home Premium operating system (64 bit). Validations and testing of GPSR Basic, GPSR Barzilai Borwein and FPC will be run on my personal computer with AMD Phenom™ $N950$ Quad-Core processor (clocked at $2.10$ GHZ) and $4.00$ GB (DDR3) memory. If the test problems are big enough, clusters at the CSCAMM will be used. $Ax$ and $A^Tx$ can be defined as function calls instead of direct matrix-vector multiplication in order to save memory allocation.

# 4 Selection of Databases

Databases are not needed for this stage of testing yet.

# 5 Validation Principles

If $\mathbf{x}$ is a solution of (7), then $\mathbf{x}$ and $\mathbf{r}(\mathbf{x})$ form an extremal pair by theorem 1.2. They satisfy (8) and (9) by theorem 1.2. We tested the GPSR Basic and GPSR Barzilai Borwein algorithms with the following setting: As it is suggested in [3] and [19], we are to validate the codes using $m = 1024$, $n = 4096$. The original signal $\mathbf{x}_*$ has $160$ non-zero entries, and these entries are randomly filled with $\pm 1$'s. Meanwhile the matrix $A$ is generated first by filling the entries $a_{ij}$ with independent samples of a standard Gaussian distribution and then orthonormalizing the rows (so that $A$ would satisfy the Restricted Isometry Property). Therefore the solution $\mathbf{x}_{\ell_1}$ from (5) would be the same as $\mathbf{x}_*$ (and as well as $\mathbf{x}_{\ell_0}$). We will take an $\tau = 0.1||A^T\mathbf{b}||_{\ell_\infty}$, and the measurement vector $\mathbf{b}$ is corrupted with noise, hence $\mathbf{b} = A\mathbf{x}_* + \zeta$, where $\zeta$ is a white Gaussian noise of a variance $\sigma^2$. $\sigma$ various from $0$, $10^{-2}$, $10^{-1}$ and $1$.

### 5.1 Validation Results

In order to validate the codes, we investigate these three quantities:

$$\begin{aligned} \text{diff}_1 &= \langle \mathbf{x}, A^T\mathbf{r}(\mathbf{x})\rangle_{\ell_2} - ||\mathbf{x}||_{\ell_1}||A^T\mathbf{r}(\mathbf{x})||_{\ell_\infty} \\ \text{diff}_2 &= ||A^T\mathbf{r}(\mathbf{x})||_{\ell_\infty} \\ J(\mathbf{x}) &= \tau||\mathbf{x}||_{\ell_1} + \frac{1}{2}||\mathbf{r}(\mathbf{x})||^2_{\ell_2} \end{aligned}$$

We hope to show that $\text{diff}_1$ and $\text{diff}_2$ are decreasing when the tolerance tolP becomes smaller:

| tolP | $\text{diff}_1$ | $\text{diff}_2$ | Num. of Iter. | $J(\mathbf{x})$ |
|---|---|---|---|---|
| $10^{-4}$ | $-2.4644e-003$ | $2.0224e-005$ | 27 | 6.7937 |
| $10^{-5}$ | $-2.2962e-004$ | $1.8994e-006$ | 33 | 6.793661 |
| $10^{-6}$ | $-2.1692e-005$ | $1.8028e-007$ | 39 | 6.7937 |
| $10^{-7}$ | $-2.0270e-006$ | $1.6894e-008$ | 45 | 6.7937 |

Table 1: Result with $\sigma = 0$ for Basic

| tolP | $\text{diff}_1$ | $\text{diff}_2$ | Num. of Iter. | $J(\mathbf{x})$ |
|---|---|---|---|---|
| $10^{-4}$ | $-2.3277e-003$ | $1.9723e-005$ | 31 | 6.7937 |
| $10^{-5}$ | $-2.0638e-004$ | $1.8571e-006$ | 37 | 6.793661 |
| $10^{-6}$ | $-2.5418e-005$ | $2.1726e-007$ | 43 | 6.7937 |
| $10^{-7}$ | $-2.2379e-006$ | $2.0232e-008$ | 49 | 6.7937 |

Table 2: Result with $\sigma = 0$ for Barzilai Borwein

| tolP | $\text{diff}_1$ | $\text{diff}_2$ | Num. of Iter. | $J(\mathbf{x})$ |
|---|---|---|---|---|
| $10^{-4}$ | $-2.0793e-003$ | $2.2583e-005$ | 28 | $6.8717e+000$ |
| $10^{-5}$ | $-1.9433e-004$ | $1.6418e-006$ | 35 | $6.871703e+000$ |
| $10^{-6}$ | $-2.1573e-005$ | $1.8249e-007$ | 41 | $6.871703e+000$ |
| $10^{-7}$ | $-2.3920e-006$ | $2.0249e-008$ | 47 | $6.871703e+000$ |

Table 3: Result with $\sigma = 10^{-2}$ for Basic

| tolP | $\text{diff}_1$ | $\text{diff}_2$ | Num. of Iter. | $J(\mathbf{x})$ |
|---|---|---|---|---|
| $10^{-4}$ | $-1.8890e-003$ | $2.0618e-005$ | 32 | $6.871703e+000$ |
| $10^{-5}$ | $-2.1573e-004$ | $2.3396e-006$ | 38 | $6.871703e+000$ |
| $10^{-6}$ | $-2.4711e-005$ | $2.6693e-007$ | 44 | $6.871703e+000$ |
| $10^{-7}$ | $-2.3235e-006$ | $1.9801e-008$ | 51 | $6.871703e+000$ |

Table 4: Result with $\sigma = 10^{-2}$ for Barzilai Borwein

| tolP | $\text{diff}_1$ | $\text{diff}_2$ | Num. of Iter. | $J(\mathbf{x})$ |
|---|---|---|---|---|
| $10^{-4}$ | $-2.5181e-003$ | $1.3017e-005$ | 65 | $1.1393e+001$ |
| $10^{-5}$ | $-2.4369e-004$ | $1.2570e-006$ | 93 | $1.1393e+001$ |
| $10^{-6}$ | $-2.6812e-005$ | $1.3817e-007$ | 121 | $1.1393e+001$ |
| $10^{-7}$ | $-2.6666e-006$ | $1.3707e-008$ | 151 | $1.1393e+001$ |

Table 5: Result with $\sigma = 10^{-1}$ for Basic

| tolP | $\text{diff}_1$ | $\text{diff}_2$ | Num. of Iter. | $J(\mathbf{x})$ |
|---|---|---|---|---|
| $10^{-4}$ | $-1.6146e-003$ | $1.3495e-005$ | 92 | $1.1393e+001$ |
| $10^{-5}$ | $-1.3726e-004$ | $1.1382e-006$ | 136 | $1.1393e+001$ |
| $10^{-6}$ | $-1.5475e-005$ | $1.2528e-007$ | 180 | $1.1393e+001$ |
| $10^{-7}$ | $-1.7440e-006$ | $1.3996e-008$ | 224 | $1.1393e+001$ |

Table 6: Result with $\sigma = 10^{-1}$ for Barzilai Borwein

| tolP | $\text{diff}_1$ | $\text{diff}_2$ | Num. of Iter. | $J(\mathbf{x})$ |
|---|---|---|---|---|
| $10^{-4}$ | $-1.0646e-002$ | $1.1406e-005$ | 349 | $1.994128e+002$ |
| $10^{-5}$ | $-1.0762e-003$ | $1.1522e-006$ | 509 | $1.994128e+002$ |
| $10^{-6}$ | $-1.0787e-004$ | $1.1550e-007$ | 673 | $1.994128e+002$ |
| $10^{-7}$ | $-5.3871e-005$ | $7.4304e-008$ | 715 | $1.994128e+002$ |

Table 7: Result with $\sigma = 1$ for Basic

| tolP | $\text{diff}_1$ | $\text{diff}_2$ | Num. of Iter. | $J(\mathbf{x})$ |
|---|---|---|---|---|
| $10^{-4}$ | $-1.0392e-002$ | $1.1516e-005$ | 327 | $1.994128e+002$ |
| $10^{-5}$ | $-1.0640e-003$ | $1.1752e-006$ | 479 | $1.994128e+002$ |
| $10^{-6}$ | $-1.0967e-004$ | $1.2108e-007$ | 635 | $1.994128e+002$ |
| $10^{-7}$ | $-1.0892e-005$ | $1.2034e-008$ | 795 | $1.1393e+001$ |

Table 8: Result with $\sigma = 1$ for Barzilai Borwein

As shown in the tables, the two quantities $\text{diff}_1$ and $\text{diff}_2$ are indeed deceasing alongside with the decreasing tolerance tolP at roughly the same speed.

## 5.2 Convergence Rate Testing for GPSR

The matrix $A$ we constructed satisfies the Restricted Isometry Property according to [3], and according to [7], $l * log(n) \leq C_M * m$; such inequality is valid for $C_M \asymp \frac{1}{23(M+1)}$, $m \leq \frac{n}{4}$, $M \geq 2$, and $n \geq 20$. Therefore $\mathbf{x}_{\ell_1}$ from (5) is the same as $\mathbf{x}_{\ell_0}$ from (3) and thus recover $\mathbf{x}_*$. According to [17], we just have to have $M = \sup_{i \neq j} \langle \mathbf{a}_i, \mathbf{a}_j \rangle_{\ell_2}$ where $\mathbf{a}_i$'s are column vectors of the matrix $A$. Then when $l \leq \frac{1}{2}(1 + \frac{1}{M})$, $\mathbf{x}_{\ell_1}$ from (5) would be the same as $\mathbf{x}_{\ell_0}$ from (3). What remains a question is that how the GPSR algorithms converges to $\mathbf{x}_{\ell_1}(\lambda)$. We will look into the details of the proof and cook up a numerical experiment for convergence rate testing.

# 6 Testing

## 6.1 Compressed Sensing Cases

In the case of Compressed Sensing testing, we can have some pre-processing information about the parameters as suggested in [18]. Let $\delta = \frac{m}{n}$ and $\gamma = \frac{l}{m}$. Then we can pick $\eta = \min(1 + 1.665(1 - \delta), 1.999)$, $\mathbf{x}^{(0)} = \eta A^T \mathbf{b}$, $\mu_1 = \theta ||\mathbf{x}^{(0)}||_\infty$ (where $0 < \theta < 1$ is a user-defined constant), set $\mu_i = \min(\mu_1 \omega^{i-1}, \bar{\mu})$ (where $\omega > 1$ and $L$ is the first integer $i$ such that $\mu_i = \bar{u}$)[7]. And we can test the solutions with the following convergence test:

$$\frac{||\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}||_{\ell_2}}{\max(||\mathbf{x}^{(k)}||_{\ell_2}, 1)} < \text{xtol} \quad \text{and} \quad \mu_i ||\mathbf{g}(\mathbf{x}^{(k)})||_{\ell_\infty} - 1 < \text{gtol} \tag{26}$$

As it was done in [18], we will takextol $= 10^{-4}$ and gtol $= 10^{-2}$. We will also try the algorithm with $A$ being either a DCT or FFT transform matrix.

### 6.1.1 Compressed Sensing Test Results

We will use $l = 160$, $m = 1024$, $n = 4086$, $A$ with entries first filled with samples of i.i.d standard Gaussian distribution and rows orthonormalized, and $\mathbf{x}_*$ would have $l$ non-zero entries (randomly picked) and filled with $\pm 1$'s (randomly assigned). As we can see from the comparison shown below that $\mathbf{x}_{\ell_2}$ from (4) does not give sparse solution, not does it give a good approximation to $\mathbf{x}_*$. Both algorithms give a very good approximation to $\mathbf{x}_*$, while the debiasing step helped even more. What the debiasing step did in this case is to push all of the non-zero entries of $\mathbf{x}_{GPSR}$ to $\pm 1$.

[7] [18] also suggests $\theta = 0.99$ and $\omega = 4$ or $\theta = 0.9$ and $\omega = 2$

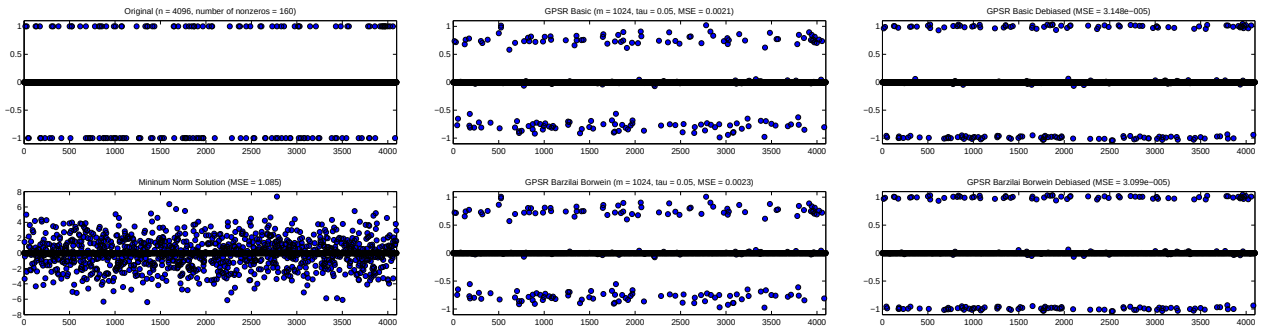



Figure 2: True Signal Vs. Various Approximations

### 6.2 Image Deconvolution, Deblurring

Plans with Image Processing experiments will be developed later after a better understanding of the vector representation of images is achieved, as well as the convolution operator.

## 7 Project Phases and Time lines

We will follow the following time lines for my project:

1. $08/29/2012$ to $10/05/2012$, Project Background Research, Project Proposal Presentation and Report.
2. $10/06/2012$ to $11/21/2012$, Implementation of the GPSR algorithm and validation by theorem 1.2.
3. $11/22/2012$ to $12/20/2012$, Implementation of the FPC algorithm and preparation for mid-year presentation and report.
4. $12/21/2012$ to $01/22/2013$, Validation of the FPC algorithm by theorem 1.2 and Convergence Rate study of GPSR.
5. $01/23/2013$ to $02/22/2013$, Implementation of the whole HD algorithm.
6. $02/23/2013$ to $03/16/2013$, Validation of the HD algorithm by theorem 1.2.
7. $03/17/2013$ to $05/20/2013$, Final Testing phase and preparation for end-of-year presentation and report.

## 8 Milestones

Here are major milestones about the project:

1. Presentation given on $10/02/2012$ and Project Proposal due on $10/05/2012$.
2. Implementation of the GPSR algorithm finished and debugged on $11/05/2012$, validation finished on $11/21/2012$.
3. Preparation given on $12/11/2012$, report due on $12/14/2012$, FPC implementation progress started, and convergence rate analysis started for GPSR.

# 9 Deliverables

MATLAB$^{TM}$ codes, presentation slides (proposal presentation, mid-year presentation, end-of-year presentation), the complete project document, test databases (if any), and test results (both in text file and/or figures) will be delivered at the end of this year long sequence.

# References


[1] J. BARZILAI AND J. BORWEIN, *Two-point step size gradient methods*, IMA Journal of Numerical Analysis, 8 (1988), pp. 141 – 148.

[2] D. P. BERTSEKAS, *Nonlinear Programming*, Athena Scientific, Boston, $2^{nd}$ ed., 1999.

[3] E. CANDES AND J. ROMBERG, *A collection of matlab routines for solving the convex optimization programs central to compressive sensing*.

[4] E. CANDES, J. ROMBERG, AND T. TAO, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Transactions on Information Theory, 52 (2004), pp. 489 – 509.

[5] ——, *Stable signal recovery from incomplete and inaccurate measurements*, Communications on Pure and Applied Mathematics, 59 (2006), pp. 1207 – 1223.

[6] E. CANDES AND T. TAO, *Decoding by linear programming*, IEEE Transactions on Information Theory, 51 (2005), pp. 4203 – 4215.

[7] ——, *Near optimal signal recovery from random projections: Universal encoding strategies?*, IEEE Transactions on Information Theory, 52 (2006), pp. 5406 – 5425.

[8] I. DAUBECHIES, M. DEFRISE, AND C. D. MOL, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Communications in Pure and Applied Mathematics, 57 (2004), pp. 1413 – 1457.

[9] D. L. DONOHO, *For most large underdetermined systems of equations, the minimal $\ell_1$-norm near-solution approximates the sparest near-solution*, technical report, Institute for Computational and Mathematical Engineering, Stanford University.

[10] ——, *For most large underdetermined systems of equations the minimal $\ell_1$-norm solution is also the sparest solution*, technical report, Institute for Computational and Mathematical Engineering, Stanford University.

[11] ——, *Unconditional bases are optimal bases for data compression and for statistical estimation*, Applied Computational Harmonic Analysis, 1 (1993), pp. 100 – 115.

[12] ——, *Sparse components of images and optimal atomic decomposition*, Constructive Approximation, 17 (2001), pp. 353 – 382.

[13] ——, *Compressed sensing*, IEEE Transactions on Information Theory, 52 (2004), pp. 1289 – 1306.

[14] D. L. DONOHO AND Y. TSAIG, *Fast solution of $\ell_1$-norm minimization problems when the solution may be sparse*, technical report, Institute for Computational and Mathematical Engineering, Stanford University.

[15] M. A. FIGUEIREDO AND R. D. NOWAK, *An em algorithm for wavelet-based image restoration*, IEEE Transactions on Image Processing, 12 (2003), pp. 906 – 916.

[16] M. A. T. FIGUEIREDO, R. D. NOWAK, AND S. J. WRIGHT, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE Journal of Selected Topics in Signal Processing, 1 (2007), pp. 586 – 597.

[17] J. J. FUCHS, *More on sparse representations in arbitrary bases*, IEEE Transactions on Information Theory, (2004), pp. 1341 – 1344.

[18] E. T. HALE, W. YIN, AND Y. ZHANG, *A fixed-point continuation method for $\ell_1$-regularized minization with applications to compressed sensing*, Technical Report TR07 - 07, CAAM.

[19] S.-J. KIM, K. KOH, M. LUSTIG, S. BOYD, AND D. GORINEVSKY, *An interior-point method for large-scale $\ell_1$-regularized least squares*, IEEE Journal of Selected Topics in Signal Processing, 1 (2007), pp. 606 – 617.

[20] B. K. NATARAJAN, *Sparse approximate solutions to linear systems*, SIAM Journal on Computing, 24 (1995), pp. 227 – 234.

[21] S. OSHER, M. BURGER, D. GOLDFARB, J. XU, AND W. YIN, *An iterative regularization method for total variation-based image restoration*, Multiscale Modeling & Simulation, 4 (2005), pp. 460 – 489.

[22] T. SERAFINI, G. ZANGHIRATI, AND L. ZANNI, *Gradient projection methods for quadratic programs and applications in training support vector machines*, Optimization Methods and Software, 20 (2004), pp. 353 – 378.

[23] E. TADMOR, S. NEZZAR, AND L. VESE, *A multiscale image representation using hierarchical (bv, $\mathcal{L}^2$) decomposition*, Multiscale Modeling & Simulation, 2 (2004), pp. 554 – 579.

[24] ——, *Multiscale hierarchical decomposition of images with applications to deblurring, denoising and segmentation*, Communications in Mathematical Sciences, 6 (2008), pp. 281 – 307.

[25] A. N. TKHONOV, *On the stability of inverse problems*, Doklady Akademii Nauk SSSR, 39 (1943), pp. 195 – 198.

[26] ——, *Solution of incorrectly formulated problems and the regularization method*, Doklady Akademii Nauk SSSR, 151 (1963), pp. 501 – 504.