SIAM J. Imaging Sciences
Vol. 4, No. 1, pp. 146–165

Gradient-Based Methods for Sparse Recovery*

William W. Hager[†], Dzung T. Phan[‡], and Hongchao Zhang[§]

Abstract. The convergence rate is analyzed for the sparse reconstruction by separable approximation (SpaRSA) algorithm for minimizing a sum $f(\mathbf{x}) + \psi(\mathbf{x})$, where f is smooth and ψ is convex, but possibly nonsmooth. It is shown that if f is convex, then the error in the objective function at iteration k is bounded by a/k for some a independent of k. Moreover, if the objective function is strongly convex, then the convergence is R-linear. An improved version of the algorithm based on a cyclic version of the BB iteration and an adaptive line search is given. The performance of the algorithm is investigated using applications in the areas of signal processing and image reconstruction.

Key words. sparse reconstruction by separable approximation, iterative shrinkage thresholding algorithm, sparse recovery, sublinear convergence, linear convergence, image reconstruction, denoising, compressed sensing, nonsmooth optimization, nonmonotone convergence, BB method

AMS subject classifications. 90C06, 90C25, 65Y20, 94A08

DOI. 10.1137/090775063

1. Introduction.

In this paper, we consider the optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad \phi(\mathbf{x}) := f(\mathbf{x}) + \psi(\mathbf{x}), \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth function, and $\psi : \mathbb{R}^n \to \mathbb{R}$ is convex. The function ψ, usually called the *regularizer* or *regularization function*, is finite for all $\mathbf{x} \in \mathbb{R}^n$, but possibly nonsmooth. An important application of (1.1), found in the signal processing literature, is the well-known ℓ_2–ℓ_1 problem (called *basis pursuit denoising* in [10])

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad \frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \tau\|\mathbf{x}\|_1, \tag{1.2}$$

where $\mathbf{A} \in \mathbb{R}^{k\times n}$ (usually $k \le n$), $\mathbf{b} \in \mathbb{R}^k, \tau \in \mathbb{R}$, $\tau \ge 0$, and $\|\cdot\|_1$ is the 1-norm.

Recently, Wright, Nowak, and Figueiredo [28] introduced the sparse reconstruction by separable approximation (SpaRSA) algorithm for solving (1.1). The algorithm has been shown to work well in practice [1, 4, 7, 28]. In [28], the authors establish global convergence of SpaRSA. In this paper, we prove an estimate of the form a/k for the error in the objective

[*] Received by the editors October 27, 2009; accepted for publication (in revised form) November 22, 2010; published electronically February 15, 2011. This material is based upon work supported by the National Science Foundation under grants 0619080 and 1016204 and by the Office of Naval Research under grant N00014-11-1-0068.
http://www.siam.org/journals/siims/4-1/77506.html

[†] Department of Mathematics, University of Florida, P.O. Box 118105, Gainesville, FL 32611-8105 (hager@ufl.edu, http://www.math.ufl.edu/~hager).

[‡] Department of Business Analytics and Mathematical Sciences, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598 (phandu@us.ibm.com).

[§] Department of Mathematics, 140 Lockett Hall, Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803-4918 (hozhang@math.lsu.edu, http://www.math.lsu.edu/~hozhang).

function when f is convex. If the objective function is strongly convex, then the convergence of the objective function and the iterates is at least R-linear. A strategy is presented for improving the performance of SpaRSA based on a cyclic Barzilai–Borwein step [11, 12, 16, 23] and an adaptive choice [18] for the reference function value in the line search. This paper concludes with a series of numerical experiments in the areas of signal processing and image reconstruction. We compare both SpaRSA related algorithms and the alternating direction method of multipliers SALSA [1].

The overall structure of the SpaRSA algorithm is closely related to that of the iterative shrinkage thresholding algorithm (ISTA) [9, 13, 15, 19, 27]. Both SpaRSA and ISTA contain parameters related to a Lipschitz constant for f; however, the rules for choosing this parameter and the line search are quite different. The line search in SpaRSA is nonmonotone, while the line search in ISTA leads to a monotone decrease in the objective function value. Nonetheless, the asymptotic convergence results we establish for SpaRSA are comparable to those for ISTA; for both methods, the error in the objective function decays like a/k when f is convex, while the convergence is at least R-linear when the objective function is strongly convex. For ISTA, this is shown by Nesterov in [22] and by Beck and Teboulle in [3]. Moreover, [3, 22] present accelerated versions of ISTA for which the error decays like a/k^2 when f is convex. Despite the similarity between the asymptotic convergence results for SpaRSA and for ISTA, and the asymptotic superiority of the a/k^2 convergence result for the accelerated ISTA schemes, it is observed in practice that SpaRSA is often much faster than either ISTA or the accelerated schemes (see [1, 4, 7, 28]).

Throughout this paper $\nabla f(\mathbf{x})$ denotes the gradient of f, a row vector. The gradient of $f(\mathbf{x})$, arranged as a column vector, is $\mathbf{g}(\mathbf{x})$. The subscript k often represents the iteration number in an algorithm, and $\mathbf{g}_k$ stands for $\mathbf{g}(\mathbf{x}_k)$. $\|\cdot\|$ denotes $\|\cdot\|_2$, the Euclidean norm. $\partial\psi(\mathbf{y})$ is the subdifferential at $\mathbf{y}$, a set of row vectors. If $\mathbf{p} \in \partial\psi(\mathbf{y})$, then

$$\psi(\mathbf{x}) \geq \psi(\mathbf{y}) + \mathbf{p}(\mathbf{x}-\mathbf{y})$$

for all $\mathbf{x} \in \mathbb{R}^n$.

2. The SpaRSA algorithm.

The SpaRSA algorithm, as presented in [28], is as follows.

Sparse Reconstruction by Separable Approximation (SpaRSA)

```
Given $\eta > 1$, $\sigma \in (0,1)$, $[\alpha_{\min}, \alpha_{\max}] \subset (0,\infty)$, and starting guess $\mathbf{x}_1$.
Set $k = 1$.
 Step 1.  Choose $\alpha_0 \in [\alpha_{\min}, \alpha_{\max}]$.
 Step 2.  Set $\alpha = \eta^j \alpha_0$, where $j \geq 0$ is the smallest integer such that

          $\phi(\mathbf{x}_{k+1}) \leq \phi_k^R - \sigma\alpha\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2$, where

          $\mathbf{x}_{k+1} = \arg\min\{\nabla f(\mathbf{x}_k)\mathbf{z} + \alpha\|\mathbf{z} - \mathbf{x}_k\|^2 + \psi(\mathbf{z}) : \mathbf{z} \in \mathbb{R}^n\}$.

 Step 3.  If a stopping criterion is satisfied, terminate the algorithm.
 Step 4.  Set $k = k + 1$ and go to Step 1.
```

In [28], the reference value ϕ_k^R is the Grippo–Lampariello–Lucidi (GLL) reference value

$\phi_k^{\max}$ [17] defined by

$$\phi_k^{\max} = \max\{\phi(\mathbf{x}_{k-j}) : 0 \le j < \min(k, M)\}. \tag{2.1}$$

In other words, at iteration k, $\phi_k^{\max}$ is the maximum of the M most recent values for the objective function. In sections 5 and 6 we introduce a different way to choose the reference value which yields better performance in the numerical experiments.

With regard to the stopping criterion mentioned in Step 3, the analysis in this paper concerns the infinite algorithm, where we allow the iterations to continue forever. Practical stopping criteria, such as those discussed in [28], are often based on the norm of the iteration difference $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|$ since $\mathbf{x}_{k+1} = \mathbf{x}_k$ if and only if $\mathbf{x}_k$ is a stationary point. In more detail, if $\mathbf{x}_{k+1} = \mathbf{x}_k$, then

$$\mathbf{0} \in \nabla f(\mathbf{x}_k) + \partial\psi(\mathbf{x}_{k+1}) = \nabla f(\mathbf{x}_{k+1}) + \partial\psi(\mathbf{x}_{k+1}),$$

which implies that $\mathbf{x}_{k+1} = \mathbf{x}_k$ is a stationary point.

The parameter α_0 in [28] was taken to be the BB parameter [2] with safeguards:

$$\alpha_0 = \alpha_k^{BB} = \arg\ \min\ \{\|\alpha\mathbf{s}_k - \mathbf{y}_k\| : \alpha_{\min} \le \alpha \le \alpha_{\max}\}, \tag{2.2}$$

where $\mathbf{s}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$ and $\mathbf{y}_k = \mathbf{g}_k - \mathbf{g}_{k-1}$. In section 6 we show that improved performance can be achieved by a cyclic scheme in which $\alpha_0 = \alpha_k^{BB}$ is reused for several subsequent iterations.

We also point out the work of Lu and Zhang [21], where a class of algorithms related to SpaRSA is analyzed. The computation of $\mathbf{x}_{k+1}$ in [21] is similar to that of SpaRSA; however, the criterion for terminating the line search is different. Lu and Zhang prove a global convergence result under a uniform continuity assumption. Decay properties for the objective function are also obtained when the derivative of f is Lipschitz continuous.

In [28] it is shown that the line search in Step 2 terminates for a finite j when f is Lipschitz continuously differentiable. Here we weaken this condition by requiring only Lipschitz continuity over a bounded set.

Proposition 2.1. *Let $\mathcal{L}$ be the level set defined by*

$$\mathcal{L} = \{\mathbf{x} \in \mathbb{R}^n : \phi(\mathbf{x}) \le \phi(\mathbf{x}_1)\}. \tag{2.3}$$

We make the following assumptions:

(A1) *The level set $\mathcal{L}$ is contained in the interior of a compact, convex set $\mathcal{K}$, and f is Lipschitz continuously differentiable on $\mathcal{K}$.*

(A2) *ψ is convex and $\psi(\mathbf{x})$ is finite for all $\mathbf{x} \in \mathbb{R}^n$.*

If $\phi(\mathbf{x}_k) \le \phi_k^R \le \phi(\mathbf{x}_1)$ in SpaRSA, then there exists $\bar{\alpha}$ with the property that

$$\phi(\mathbf{x}_{k+1}) \le \phi_k^R - \sigma\alpha\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2$$

whenever $\alpha \ge \bar{\alpha}$, where

$$\mathbf{x}_{k+1} = \arg\ \min\{\nabla f(\mathbf{x}_k)\mathbf{z} + \alpha\|\mathbf{z} - \mathbf{x}_k\|^2 + \psi(\mathbf{z}) : \mathbf{z} \in \mathbb{R}^n\}.$$

Moreover, if $\phi(\mathbf{x}_k) \le \phi_k^R \le \phi(\mathbf{x}_1)$ for all k, then there exists a constant $\bar{\beta}$ such that $\alpha_k \le \bar{\beta} < \infty$ for all k.

Proof. Let Φ_k be defined by

$$\Phi_k(\mathbf{z}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)(\mathbf{z} - \mathbf{x}_k) + \alpha\|\mathbf{z} - \mathbf{x}_k\|^2 + \psi(\mathbf{z}), \tag{2.4}$$

where $\alpha \geq 0$. Since Φ_k is a strongly convex quadratic, its level sets are compact. Let $\mathbf{x}_{k+1}$ denote the unique minimizer of Φ_k. Expanding in a Taylor series around $\mathbf{x}_k$, we have

$$\begin{aligned}\Phi_k(\mathbf{x}_{k+1}) &= f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k) + \alpha\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \psi(\mathbf{x}_{k+1}) \\ &\leq \Phi_k(\mathbf{x}_k) = f(\mathbf{x}_k) + \psi(\mathbf{x}_k).\end{aligned}$$

This is rearranged to obtain

$$\begin{aligned}\alpha\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 &\leq \nabla f(\mathbf{x}_k)(\mathbf{x}_k - \mathbf{x}_{k+1}) + \psi(\mathbf{x}_k) - \psi(\mathbf{x}_{k+1}) \\ &\leq \nabla f(\mathbf{x}_k)(\mathbf{x}_k - \mathbf{x}_{k+1}) + \mathbf{p}_k(\mathbf{x}_k - \mathbf{x}_{k+1}),\end{aligned}$$

where $\mathbf{p}_k \in \partial\psi(\mathbf{x}_k)$. Taking norms yields

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq (\|\mathbf{g}_k\| + \|\mathbf{p}_k\|)/\alpha. \tag{2.5}$$

By Theorem 23.4 and Corollary 24.5.1 in [24] and by the compactness of $\mathcal{L}$, there exists a constant c, independent of $\mathbf{x}_k \in \mathcal{L}$, such that $\|\mathbf{g}_k\| + \|\mathbf{p}_k\| \leq c$. Consequently, we have

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq c/\alpha.$$

Since $\mathcal{K}$ is compact and $\mathcal{L}$ lies in the interior of $\mathcal{K}$, the distance δ from $\mathcal{L}$ to the boundary of $\mathcal{K}$ is positive. Choose $\beta \in (0, \infty)$ so that $c/\beta \leq \delta$. Hence, when $\alpha \geq \beta$, $\mathbf{x}_{k+1} \in \mathcal{K}$ since $\mathbf{x}_k \in \mathcal{L}$.

Let Λ denote the Lipschitz constant for f on $\mathcal{K}$, and suppose that $\alpha \geq \beta$. Since $\mathbf{x}_k \in \mathcal{L} \subset \mathcal{K}$ and $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq \delta$, we have $\mathbf{x}_{k+1} \in \mathcal{K}$. Moreover, due to the convexity of $\mathcal{K}$, the line segment connecting $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ lies in $\mathcal{K}$. Proceeding as in [28], a Taylor expansion around $\mathbf{x}_k$ yields

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k) + .5\Lambda\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2.$$

Adding $\psi(\mathbf{x}_{k+1})$ to both sides, we have

$$\begin{aligned}\phi(\mathbf{x}_{k+1}) &\leq \Phi_k(\mathbf{x}_{k+1}) + (.5\Lambda - \alpha)\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\ &\leq \Phi_k(\mathbf{x}_k) + (.5\Lambda - \alpha)\, \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\ &= \phi(\mathbf{x}_k) + (.5\Lambda - \alpha)\, \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\ &\leq \phi_k^R + (.5\Lambda - \alpha)\, \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \quad \text{since } \phi(\mathbf{x}_k) \leq \phi_k^R \\ &\leq \phi_k^R - \sigma\alpha\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \quad \text{if } .5\Lambda - \alpha \leq -\sigma\alpha.\end{aligned} \tag{2.6}$$

Hence, the proposition holds with

$$\bar{\alpha} = \max\left\{\beta, \frac{\Lambda}{2(1-\sigma)}\right\}.$$

Let j be the nonnegative integer generated in Step 2 of SpaRSA at iteration k. If $\alpha_0 \geq \bar{\alpha}$, then $\alpha_k = \alpha_0 \leq \alpha_{\max}$. If $\alpha_0 < \bar{\alpha}$, then $\eta^{j-1}\alpha_0 \leq \bar{\alpha}$. Hence, $\alpha_k = \eta(\eta^{j-1}\alpha_0) \leq \eta\bar{\alpha}$, and we have

$$\alpha_k \leq \bar{\beta} := \max\{\alpha_{\max}, \eta\bar{\alpha}\}. \qquad \blacksquare$$

Remark 1. Suppose $\phi_k^R \in [\phi(\mathbf{x}_k), \phi(\mathbf{x}_1)]$. In Step 2 of SpaRSA, $\mathbf{x}_{k+1}$ is chosen so that $\phi(\mathbf{x}_{k+1}) \le \phi_k^R$. Since the interval $[\phi(\mathbf{x}_{k+1}), \phi(\mathbf{x}_1)]$ contains $[\phi_k^R, \phi(\mathbf{x}_1)]$, there exists ϕ_{k+1}^R which satisfies the assumption of Proposition 2.1 at step $k+1$.

Remark 2. We now show that the GLL reference value $\phi_k^{\max}$ satisfies the condition $\phi(\mathbf{x}_k) \le \phi_k^R \le \phi(\mathbf{x}_1)$ of Proposition 2.1 for each k. The condition $\phi_k^{\max} \ge \phi(\mathbf{x}_k)$ is a trivial consequence of the definition of $\phi_k^{\max}$. Also, by the definition, we have $\phi_1^{\max} = \phi(\mathbf{x}_1)$. For $k \ge 1$, $\phi(\mathbf{x}_{k+1}) \le \phi_k^{\max}$ according to Step 2 of SpaRSA. Hence, $\phi_k^{\max}$ is a decreasing function of k. In particular, $\phi_k^{\max} \le \phi_1^{\max} = \phi(\mathbf{x}_1)$.

3. Convergence estimate for convex functions.

In this section we give a sublinear convergence estimate for the error in the objective function value $\phi(\mathbf{x}_k)$, assuming f is convex and the assumptions of Proposition 2.1 hold.

By (A1) and (A2), (1.1) has a solution $\mathbf{x}^* \in \mathcal{L}$ and an associated objective function value $\phi^* := \phi(\mathbf{x}^*)$. The convergence of the objective function values to ϕ^* is a consequence of the analysis in [28].

Lemma 3.1. *If* (A1) *and* (A2) *hold, f is convex, and $\phi_k^R = \phi_k^{\max}$ for every k, then*

$$\lim_{k\to\infty} \phi(\mathbf{x}_k) = \phi^*.$$

Proof. By [28, Lemma 4], the objective function values $\phi(\mathbf{x}_k)$ approach a limit denoted $\bar{\phi}$. By [28, Theorem 1], all accumulation points of the iterates $\mathbf{x}_k$ are stationary points. An accumulation point exists since $\mathcal{K}$ is compact and the iterates are all contained in $\mathcal{L} \subset \mathcal{K}$, as shown in Remark 2. Since f and ψ are both convex, a stationary point is a global minimizer of ϕ. Hence, $\bar{\phi} = \phi^*$. ■

Our sublinear convergence result is the following.

Theorem 3.2. *If* (A1) *and* (A2) *hold, f is convex, and $\phi_k^R = \phi_k^{\max}$ for all k, then there exists a constant a such that*

$$\phi(\mathbf{x}_k) - \phi^* \le \frac{a}{k}$$

for all k, where ϕ^* is the optimal objective function value for (1.1).

Proof. By (2.6) with $k+1$ replaced by k, we have

$$\phi(\mathbf{x}_k) \le \Phi_{k-1}(\mathbf{x}_k) + b_0\|\mathbf{s}_k\|^2, \quad b_0 = .5\Lambda, \tag{3.1}$$

where $\mathbf{s}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$ and Φ_k is defined in (2.4). Since $\mathbf{x}_k$ minimizes Φ_{k-1} and f is convex, it follows that

$$\begin{aligned} \Phi_{k-1}(\mathbf{x}_k) &= \min_{\mathbf{z}\in\mathbb{R}^n} \{f(\mathbf{x}_{k-1}) + \nabla f(\mathbf{x}_{k-1})(\mathbf{z} - \mathbf{x}_{k-1}) + \alpha_{k-1}\|\mathbf{z} - \mathbf{x}_{k-1}\|^2 + \psi(\mathbf{z})\} \\ &\le \min\{f(\mathbf{z}) + \psi(\mathbf{z}) + \alpha_{k-1}\|\mathbf{z} - \mathbf{x}_{k-1}\|^2 : \mathbf{z} \in \mathbb{R}^n\} \\ &= \min\{\phi(\mathbf{z}) + \alpha_{k-1}\|\mathbf{z} - \mathbf{x}_{k-1}\|^2 : \mathbf{z} \in \mathbb{R}^n\}, \end{aligned} \tag{3.2}$$

where α_{k-1} is the terminating value of α at step $k-1$. Combining (3.1) and (3.2) gives

$$\phi(\mathbf{x}_k) \le \min\{\phi(\mathbf{z}) + \bar{\beta}\|\mathbf{z} - \mathbf{x}_{k-1}\|^2 : \mathbf{z} \in \mathbb{R}^n\} + b_0\|\mathbf{s}_k\|^2, \tag{3.3}$$

where $\bar{\beta}$ is the upper bound for α_{k-1} given in Proposition 2.1. We take

$$\mathbf{z} = (1-\lambda)\mathbf{x}_{k-1} + \lambda\mathbf{x}^*,$$

where $\lambda \in [0,1]$ and $\mathbf{x}^*$ is a solution of (1.1). By the convexity of ϕ, we have

$$\begin{aligned}\min_{\mathbf{z}\in\mathbb{R}^n} \phi(\mathbf{z}) + \bar{\beta}\|\mathbf{z}-\mathbf{x}_{k-1}\|^2 &\le \phi((1-\lambda)\mathbf{x}_{k-1} + \lambda\mathbf{x}^*) + \bar{\beta}\lambda^2\|\mathbf{x}_{k-1}-\mathbf{x}^*\|^2 \\ &\le (1-\lambda)\phi(\mathbf{x}_{k-1}) + \lambda\phi^* + \bar{\beta}\lambda^2\|\mathbf{x}_{k-1}-\mathbf{x}^*\|^2 \\ &= (1-\lambda)\phi(\mathbf{x}_{k-1}) + \lambda\phi^* + \beta_k\lambda^2,\end{aligned}$$

where $\beta_k := \bar{\beta}\|\mathbf{x}_{k-1}-\mathbf{x}^*\|^2$. Combining this with (3.3) yields

$$\begin{aligned}\phi(\mathbf{x}_k) &\le (1-\lambda)\phi(\mathbf{x}_{k-1}) + \lambda\phi^* + \beta_k\lambda^2 + b_0\|\mathbf{s}_k\|^2 \\ &\le (1-\lambda)\phi^R_{k-1} + \lambda\phi^* + \beta_k\lambda^2 + b_0\|\mathbf{s}_k\|^2\end{aligned} \tag{3.4}$$

for any $\lambda \in [0,1]$. Define

$$\phi_i = \max\{\phi(\mathbf{x}_k) : (i-1)M < k \le iM\} = \phi^R_{iM} = \phi^{\max}_{iM}, \tag{3.5}$$

and let k_i denote the index k where the maximum is attained in (3.5). Since $\phi(\mathbf{x}_{k+1}) \le \phi^R_k$ in Step 2 of SpaRSA, it follows that $\phi^R_k = \phi^{\max}_k$ is a nonincreasing function of k. By (3.4) with $k = k_i$ and by the monotonicity of ϕ^R_k, we have

$$\phi_i \le (1-\lambda)\phi_{i-1} + \lambda\phi^* + \beta_{k_i}\lambda^2 + b_0\|\mathbf{s}_{k_i}\|^2 \tag{3.6}$$

for any $\lambda \in [0,1]$. Since both $\mathbf{x}_{k-1}$ and $\mathbf{x}^*$ lie in $\mathcal{L}$, it follows that

$$\beta_k = \bar{\beta}\|\mathbf{x}_{k-1}-\mathbf{x}^*\|^2 \le \bar{\beta}(\text{diameter of } \mathcal{L})^2 := b_2 < \infty. \tag{3.7}$$

Step 2 of SpaRSA implies that

$$\|\mathbf{s}_k\|^2 \le (\phi^R_{k-1} - \phi(\mathbf{x}_k))/b_1,$$

where $b_1 = \sigma\alpha_{\min}$. We take $k = k_i$ and again exploit the monotonicity of ϕ^R_k to obtain

$$\|\mathbf{s}_{k_i}\|^2 \le (\phi_{i-1} - \phi_i)/b_1. \tag{3.8}$$

Combining (3.6)–(3.8) gives

$$\phi_i \le (1-\lambda)\phi_{i-1} + \lambda\phi^* + b_2\lambda^2 + b_3(\phi_{i-1} - \phi_i), \quad b_3 = b_0/b_1, \tag{3.9}$$

for every $\lambda \in [0,1]$, The minimum on the right-hand side is attained with the choice

$$\lambda = \min\left\{1, \frac{\phi_{i-1} - \phi^*}{2b_2}\right\}. \tag{3.10}$$

As a consequence of Lemma 3.1, ϕ_{i-1} converges to ϕ^*. Hence, the minimizing λ also approaches 0 as i tends to ∞. Choose k large enough that the minimizing λ is less than 1. It follows from (3.9) that when this minimizing choice of λ is less than 1, we have

$$\phi_i \le \phi_{i-1} - \frac{(\phi_{i-1} - \phi^*)^2}{4b_2} + b_3(\phi_{i-1} - \phi_i). \tag{3.11}$$

Define $e_i = \phi_i - \phi^*$. Subtracting ϕ^* from each side of (3.11) gives

$$\begin{aligned} e_i &\le e_{i-1} - e_{i-1}^2/(4b_2) + b_3(e_{i-1} - e_i) \\ &= (1 + b_3)e_{i-1} - e_{i-1}^2/(4b_2) - b_3 e_i. \end{aligned}$$

We arrange this to obtain

$$e_i \le e_{i-1} - b_4 e_{i-1}^2, \quad \text{where } b_4 = \frac{1}{4b_2(1 + b_3)}. \tag{3.12}$$

By (3.12), $e_i \le e_{i-1}$, which implies that

$$e_i \le e_{i-1} - b_4 e_{i-1} e_i \quad \text{or} \quad e_i \le \frac{e_{i-1}}{1 + b_4 e_{i-1}}.$$

We form the reciprocal of this last inequality to obtain

$$\frac{1}{e_i} \ge \frac{1}{e_{i-1}} + b_4.$$

Applying this inequality recursively gives

$$\frac{1}{e_i} \ge \frac{1}{e_{i_0}} + (i - i_0)b_4 \quad \text{or} \quad e_i \le \frac{e_{i_0}}{1 + (i - i_0)b_4 e_{i_0}} \le \frac{1}{(i - i_0)b_4}, \quad i > i_0,$$

where i_0 is chosen large enough to ensure that the minimizing λ in (3.10) is less than 1 for all $i \ge i_0$. If $i \ge 2i_0$, then we have

$$e_i \le \frac{1}{(i - i_0)b_4} = \frac{1}{i(1 - i_0/i)b_4} \le \frac{2}{ib_4}.$$

Choose $b_5 > 2/b_4$ large enough so that $e_i \le b_5/i$ for all integers $i \in [1, 2i_0]$. Since there are a finite number of $i \in [1, 2i_0]$, there exists a finite b_5 with this property. Then, for all i, we have

$$e_i \le b_5/i.$$

Suppose that $k \in ((i-1)M, iM]$. Since $i \ge k/M$, we have

$$\phi(\mathbf{x}_k) - \phi^* \le e_i = \frac{b_5}{i} \le \frac{b_5}{k/M} = \frac{b_5 M}{k}.$$

The proof is completed by taking $a = b_5 M$. ■

4. Convergence estimate for strongly convex functions. In this section we prove that SpaRSA converges R-linearly when f is a convex function and ϕ satisfies

$$\phi(\mathbf{y}) \ge \phi(\mathbf{x}^*) + \mu\|\mathbf{y} - \mathbf{x}^*\|^2 \tag{4.1}$$

for all $\mathbf{y} \in \mathbb{R}^n$, where $\mu > 0$. Hence, $\mathbf{x}^*$ is a unique minimizer of ϕ. For example, if f is a strongly convex function, then (4.1) holds.

Theorem 4.1. *If* (A1) *and* (A2) *hold, f is convex, ϕ satisfies* (4.1)*, and $\phi_k^R = \phi_k^{\max}$ for every k, then there exist constants $\theta \in (0,1)$ and c such that*

$$\phi(\mathbf{x}_k) - \phi^* \le c\theta^k(\phi(\mathbf{x}_1) - \phi^*) \tag{4.2}$$

for every k.

Proof. As in the proof of Theorem 3.2, we define

$$\phi_i = \max\{\phi(\mathbf{x}_k) : (i-1)M < k \le iM\} = \phi_{iM}^{\max},$$

and we let k_i denote the index k where the maximum is attained. We will show that there exists $\gamma \in (0,1)$ such that

$$\phi_i - \phi^* \le \gamma(\phi_{i-1} - \phi^*). \tag{4.3}$$

Let c_1 be chosen to satisfy the inequality

$$0 < c_1 < \min\left\{\frac{1}{2b_0}, \frac{\mu}{4b_0\bar{\beta}}\right\}. \tag{4.4}$$

Here $b_0 = .5\Lambda$, where Λ is the Lipschitz constant for f, μ is the strong convexity constant in (4.1), and $\bar{\beta}$ is the upper bound for the SpaRSA step size α_k given in Proposition 2.1. Let $\mathbf{s}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$ be the iteration difference. We consider two cases.

Case 1. $\|\mathbf{s}_{k_i}\|^2 \ge c_1(\phi_{i-1} - \phi^*)$.

By (3.8), we have

$$c_1(\phi_{i-1} - \phi^*) \le \|\mathbf{s}_{k_i}\|^2 \le (\phi_{i-1} - \phi_i)/b_1$$

for some constant b_1. This can be rearranged to obtain

$$\phi_i - \phi^* \le (1 - b_1c_1)(\phi_{i-1} - \phi^*),$$

which yields (4.3).

Case 2. $\|\mathbf{s}_{k_i}\|^2 < c_1(\phi_{i-1} - \phi^*)$.

For $k \in ((i-1)M, iM]$, we have

$$\begin{aligned}\beta_k := \bar{\beta}\|\mathbf{x}_{k-1} - \mathbf{x}^*\|^2 &\le \frac{\bar{\beta}}{\mu}(\phi(\mathbf{x}_{k-1}) - \phi^*) \le \frac{\bar{\beta}}{\mu}(\phi_{k-1}^R - \phi^*)\\ &\le \frac{\bar{\beta}}{\mu}(\phi_{(i-1)M}^R - \phi^*) = b_6(\phi_{i-1} - \phi^*), \quad b_6 = \frac{\bar{\beta}}{\mu}.\end{aligned}$$

The first inequality is due to (4.1), the second inequality is due to the assumption that $\phi_{k-1}^R = \phi_{k-1}^{\max}$, and the last inequality is due to ϕ_k^R being monotone decreasing. Making the choice $k = k_i$, we obtain

$$\beta_{k_i} = \bar{\beta}\|\mathbf{x}_{k_i-1} - \mathbf{x}^*\|^2 \le b_6(\phi_{i-1} - \phi^*). \tag{4.5}$$

Inserting into (3.6) the bound (4.5) and the Case 2 requirement $\|\mathbf{s}_{k_i}\|^2 < c_1(\phi_{i-1} - \phi^*)$ yields

$$\phi_i \le (1-\lambda)\phi_{i-1} + \lambda\phi^* + b_6(\phi_{i-1} - \phi^*)\lambda^2 + b_0 c_1(\phi_{i-1} - \phi^*)$$

for all $\lambda \in [0, 1]$. Subtract ϕ^* from each side to obtain

$$e_i \le [1 + b_0 c_1 - \lambda + b_6\lambda^2] e_{i-1} \tag{4.6}$$

for all $\lambda \in [0, 1]$, where $e_i = \phi(\mathbf{x}_k) - \phi^*$.

The $\lambda \in [0, 1]$ which minimizes the coefficient of e_{i-1} in (4.6) is

$$\lambda = \min\left\{1, \frac{1}{2b_6}\right\}.$$

If the minimizing λ is 1, then $b_6 \le 1/2$ and the minimizing coefficient in (4.6) is

$$\gamma = b_0 c_1 + b_6 \le b_0 c_1 + 1/2 < 1$$

since $c_1 < 1/(2b_0)$ by (4.4). On the other hand, if the minimizing λ is less than 1, then $b_6 > 1/2$ and the minimizing coefficient in (4.6) is

$$\gamma = 1 + b_0 c_1 - \frac{1}{4b_6} < 1$$

since $1/(4b_6) = \mu/(4\bar{\beta}) > b_0 c_1$ by (4.4). This completes the proof of (4.3).

For $k \in ((i-1)M, iM]$, we have $i \ge k/M$ and

$$\phi(\mathbf{x}_k) - \phi^* = e_i \le \gamma^{i-1} e_1 \le \frac{1}{\gamma}\left(\gamma^{1/M}\right)^k (\phi(\mathbf{x}_1) - \phi^*).$$

Hence, (4.2) holds with $c = 1/\gamma$ and $\theta = \gamma^{1/M}$. This completes the proof. ■

Remark 3. The condition (4.1) when combined with (4.2) shows that the iterates $\mathbf{x}_k$ converge R-linearly to $\mathbf{x}^*$.

5. More general reference function values.

The GLL reference function value $\phi_k^{\max}$, defined in (2.1), often leads to greater efficiency when $M > 1$, when compared to the monotone choice $M = 1$. In practice, it is found that even more flexibility in the reference function value can further accelerate convergence. In [18] we prove convergence of the nonmonotone gradient projection method whenever the reference function ϕ_k^R satisfies the following conditions:

(R1) $\phi_1^R = \phi(\mathbf{x}_1)$.
(R2) $\phi(\mathbf{x}_k) \le \phi_k^R \le \max\{\phi_{k-1}^R, \phi_k^{\max}\}$ for each $k > 1$.
(R3) $\phi_k^R \le \phi_k^{\max}$ infinitely often.

In this section, we give convergence results for SpaRSA whenever the reference function value satisfies (R1)–(R3). In the first convergence result which follows, convexity of f is not required.

Theorem 5.1. *If* (A1) *and* (A2) *hold and the reference function value* ϕ_k^R *satisfies* (R1)–(R3)*, then the iterates* $\mathbf{x}_k$ *of SpaRSA have a subsequence converging to a limit* $\bar{\mathbf{x}}$ *satisfying* $\mathbf{0} \in \partial\phi(\bar{\mathbf{x}})$.

Proof. We first apply Proposition 2.1 to show that Step 2 of SpaRSA is fulfilled for some choice of j. By (R2), $\phi(\mathbf{x}_k) \le \phi_k^R$ for each k. Hence, to apply Proposition 2.1, we must show that $\phi_k^R \le \phi(\mathbf{x}_1)$ for each k. This holds for $k = 1$ by (R1). Also, for $k = 1$, we have $\phi_1^{\max} = \phi(\mathbf{x}_1)$. Proceeding by induction, suppose that $\phi_i^R \le \phi(\mathbf{x}_1)$ and $\phi_i^{\max} \le \phi(\mathbf{x}_1)$ for $i = 1, 2, \ldots, k$. By Proposition 2.1, at iteration k, there exists a finite j such that Step 2 of SpaRSA is fulfilled and hence

$$\phi(\mathbf{x}_{k+1}) \le \phi_k^R \le \phi(\mathbf{x}_1).$$

It follows that $\phi_{k+1}^{\max} \le \phi(\mathbf{x}_1)$ and $\phi_{k+1}^R \le \max\{\phi_k^R, \phi_{k+1}^{\max}\} \le \phi(\mathbf{x}_1)$. This completes the induction step, and hence, by Proposition 2.1, it follows that in every iteration, Step 2 of SpaRSA is fulfilled for a finite j.

By Step 2 of SpaRSA, we have

$$\phi(\mathbf{x}_k) \le \phi_{k-1}^R - \sigma\alpha_{\min}\|\mathbf{s}_k\|^2,$$

where $\mathbf{s}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$. In the third paragraph of the proof of Theorem 2.2 in [18], it is shown that when an inequality of this form is satisfied for a reference function value satisfying (R1)–(R3),

$$\liminf_{k\to\infty} \|\mathbf{s}_k\| = 0.$$

Let k_i denote a strictly increasing sequence with the property that $\mathbf{s}_{k_i}$ tends to $\mathbf{0}$ and $\mathbf{x}_{k_i}$ approaches a limit denoted $\bar{\mathbf{x}}$. That is,

$$\lim_{i\to\infty} \mathbf{s}_{k_i} = 0 \quad \text{and} \quad \lim_{i\to\infty} \mathbf{x}_{k_i} = \bar{\mathbf{x}}.$$

Since $\mathbf{s}_{k_i}$ tends to $\mathbf{0}$, it follows that $\mathbf{x}_{k_i-1}$ also approaches $\bar{\mathbf{x}}$. By the first-order optimality conditions for $\mathbf{x}_{k_i}$, we have

$$\mathbf{0} \in \nabla f(\mathbf{x}_{k_i-1}) + 2\alpha_{k_i}(\mathbf{x}_{k_i} - \mathbf{x}_{k_i-1}) + \partial\psi(\mathbf{x}_{k_i}), \tag{5.1}$$

where α_{k_i} denotes the value of α in Step 2 of SpaRSA associated with $\mathbf{x}_{k_i}$. Again, by Proposition 2.1, we have the uniform bound $\alpha_{k_i} \le \bar{\beta}$. Taking the limit as i tends to ∞, it follows from Corollary 24.5.1 in [24] that

$$\mathbf{0} \in \nabla f(\bar{\mathbf{x}}) + \partial\psi(\bar{\mathbf{x}}).$$

This completes the proof. ■

With a small change in (R3), we obtain either sublinear or linear convergence of the entire iteration sequence.

Theorem 5.2. *Suppose that* (A1) *and* (A2) *hold, f is convex, the reference function value ϕ_k^R satisfies* (R1) *and* (R2)*, and there is $L > 0$ with the property that for each k,*

$$\phi_j^R \le \phi_j^{\max} \quad \text{for some } j \in [k, k+L). \tag{5.2}$$

Then there exists a constant a such that

$$\phi(\mathbf{x}_k) - \phi^* \le \frac{a}{k}$$

for all k. Moreover, if ϕ satisfies the strong convexity condition (4.1)*, then there exist $\theta \in (0,1)$ and c such that*

$$\phi(\mathbf{x}_k) - \phi^* \le c\theta^k(\phi(\mathbf{x}_1) - \phi^*)$$

for every k.

Proof. Let k_i, $i = 1, 2, \ldots$, denote an increasing sequence of integers with the property that $\phi_j^R \le \phi_j^{\max}$ for $j = k_i$ and $\phi_j^R \le \phi_{j-1}^R$ when $k_i < j < k_{i+1}$. Such a sequence exists since $\phi_k^R \le \max\{\phi_{k-1}^R, \phi_k^{\max}\}$ for each k and (5.2) holds. Moreover, $k_{i+1} - k_i \le L$. Hence, we have

$$\phi_j^R \le \phi_{k_i}^R \le \phi_{k_i}^{\max} \quad \text{when } k_i \le j < k_{i+1}. \tag{5.3}$$

Let us define

$$\phi_j^{\max +} = \max\{\phi(\mathbf{x}_{j-i}) : 0 \le i < \min(j, M+L)\}.$$

Note that the memory associated with $\phi_j^{\max}$ is M while the memory associated with $\phi_j^{\max +}$ is $M + L$. Given j, choose k_i such that $j \in [k_i, k_{i+1})$. Since $j - k_i < L$, the set of function values maximized to obtain $\phi_{k_i}^{\max}$ is contained in the set of function values maximized to obtain $\phi_j^{\max +}$ and we have

$$\phi_{k_i}^{\max} \le \phi_j^{\max +}. \tag{5.4}$$

Combining (5.3) and (5.4) yields $\phi_j^R \le \phi_j^{\max +}$ for each j. In Step 2 of SpaRSA, the iterates are chosen to satisfy the condition

$$\phi(\mathbf{x}_{k+1}) \le \phi_k^R - \sigma\alpha\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2.$$

It follows that

$$\phi(\mathbf{x}_{k+1}) \le \phi_k^{\max +} - \sigma\alpha\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2.$$

Hence, the iterates also satisfy the GLL condition, but with memory of length $M + L$ instead of M. By Theorem 3.2, the iterates converge at least sublinearly. Moreover, if the strong convexity condition (4.1) holds, then the convergence is R-linear by Theorem 4.1. ■

6. Computational experiments.

In this section, we compare the performance of three methods: SpaRSA with the GLL reference function value $\phi_k^{\max}$ and the BB choice for α_0 in SpaRSA; the alternating direction method of multipliers SALSA [1]; and our adaptive implementation of SpaRSA based on the reference function value ϕ_k^R given in the appendix of [18] and a cyclic BB choice for α_0. We call this implementation Adaptive SpaRSA. This adaptive choice for ϕ_k^R satisfies (R1)–(R3), which ensures convergence in accordance with

Theorem 5.1. By a cyclic choice for the BB parameter (see [11, 12, 16, 23]), we mean that $\alpha_0 = \alpha_k^{BB}$ is reused for several iterations. More precisely, for some integer $m \geq 1$ (the cycle length), and for all $k \in ((i-1)m, im]$, the value of α_0 at iteration k is given by

$$(\alpha_0)_k = \alpha_{(i-1)m+1}^{BB}.$$

We now explain the basic idea underlying the choice of reference function value given in [18]. The goal is to set $\phi_k^R = \phi_{k-1}^R$ as often as possible and resort to setting $\phi_k^R = \phi_k^{\max}$ only if the objective function value is not decaying adequately. More precisely, suppose that we would like to see a decay of size $\Delta > 0$ in function value within L iterations ($L = 3$ in the experiments). If, after L iterations, the objective function value does not decay by Δ, then we would set $\phi_k^R = \phi_k^{\max}$; otherwise $\phi_k^R = \phi_{k-1}^R$. Assuming ϕ is bounded from below, this process will set $\phi_k^R = \phi_k^{\max}$ infinitely often; consequently, condition (R3) is satisfied. Condition (R2) is satisfied since ϕ_k^R is either ϕ_{k-1}^R or $\phi_k^{\max}$. See the appendix of [18] for full details. In the numerical experiments, we simply check that the decay in the function value over L iterations is strictly positive. In finite precision arithmetic, there exists $\delta > 0$ such $y - x > \delta$ whenever $x < y$.

The test problems are associated with applications in the areas of signal processing and image reconstruction. All experiments were carried out on a PC using MATLAB 7.6 with an AMD Athlon 64 X2 dual core 3 Ghz processor and 3GB of memory running Windows Vista. Version 2.0 of SpaRSA was obtained from Mário Figueiredo's webpage (http://www.lx.it.pt/~mtf). Version 1.0 of SALSA was obtained from Manya Afonso's webpage (http://cascais.lx.it.pt/~mafonso). The codes were run with default parameters. Adaptive SpaRSA was written in MATLAB with the following parameter values:

$$\alpha_{\min} = 10^{-30}, \quad \alpha_{\max} = 10^{30}, \quad \eta = 5, \quad \sigma = 10^{-4}, \quad M = 10.$$

The test problems, such as the basis pursuit denoising problem (1.2), involve a parameter τ. The choice of the cycle length was based on the value of τ:

$$m = 1 \text{ if } \tau > 10^{-2}; \quad \text{otherwise } m = 3.$$

As τ approaches zero, the optimization problem becomes more ill conditioned and the convergence speed improves when the cycle length is increased.

The stopping condition for both SpaRSA and Adaptive SpaRSA was

$$\alpha_k \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_\infty \leq \epsilon,$$

where α_k denotes the final value for α in Step 2 of SpaRSA, $\|\cdot\|_\infty$ is the max-norm, and ϵ is the error tolerance. This termination condition is suggested by Vandenberghe in [26]. As pointed out earlier, $\mathbf{x}_k$ is a stationary point when $\mathbf{x}_{k+1} = \mathbf{x}_k$. For other stopping criteria, see [19] or [28]. In the following tables, "Ax" denotes the number of times that a vector is multiplied by $\mathbf{A}$ or $\mathbf{A}^\mathsf{T}$, "cpu" is the CPU time in seconds, and "Obj" is the objective function value.

6.1. ℓ_2–ℓ_1 problems. We compare the performance of Adaptive SpaRSA with SpaRSA by solving ℓ_2–ℓ_1 problems of form (1.2) using the randomly generated data introduced in [20, 28]. The matrix $\mathbf{A}$ is a random $k \times n$ matrix, with $k = 2^8$ and $n = 2^{10}$. The elements of $\mathbf{A}$ are chosen from a Gaussian distribution with mean zero and variance $1/(2n)$. The observed vector is $\mathbf{b} = \mathbf{A}\mathbf{x}_{true} + \mathbf{n}$, where the noise $\mathbf{n}$ is sampled from a Gaussian distribution with mean zero and variance 10^{-4}. $\mathbf{x}_{true}$ is a vector with 160 randomly placed ±1 spikes with zeros in the remaining elements. This is a typical sparse signal recovery problem which often arises in compressed sensing [14]. We solved the problem (1.2) corresponding to the error tolerance $\epsilon = 10^{-5}$ with different regularization parameters τ between 10^{-1} and 10^{-5}. Table 1 reports the average CPU times (seconds) and the number of matrix-vector multiplications over 10 runs for SpaRSA-based algorithms. The "/c" entries in the table represent an implementation based on the continuation method presented in [19]. The SpaRSA entries refer to the original version of the algorithm with a GLL line search; in SpaRSA+ we replace the GLL line search by our line search based on the reference function value described in section 5; finally, Adaptive refers to SpaRSA with the new line search and with the cyclic BB method. These results show that both the new line search and the cyclic BB method contributed to improvements in performance. Also, the continuation method led to a further improvement in performance.

Figure 1 plots error versus the number of matrix-vector multiplications for $\tau = 10^{-4}$ and the implementation without continuation. When the error is large, both algorithms have the same performance. As the error tolerance decreases, the performance of the adaptive algorithm is significantly better than the original implementation.

Table 1

Average over 10 runs for ℓ_2–ℓ_1 problems.

τ	1e-1		1e-2		1e-3		1e-4		1e-5	
	Ax	cpu	Ax	cpu	Ax	cpu	Ax	cpu	Ax	cpu
SpaRSA	67.1	.08	892.3	.75	3241.0	2.54	8992.9	6.99	6295.3	4.92
SpaRSA+	67.0	.08	717.3	.57	2936.9	2.27	7980.7	6.25	5692.4	4.38
Adaptive	67.0	.08	641.4	.51	1878.8	1.46	4686.5	3.72	2931.6	2.25
SpaRSA/c	67.1	.08	809.2	.63	2265.8	1.81	698.7	.53	475.7	.35
SpaRSA+/c	67.0	.08	686.9	.54	1952.6	1.52	653.4	.50	455.1	.33
Adaptive/c	67.0	.08	602.3	.46	1355.4	1.08	604.2	.46	442.9	.33

6.2. Image deblurring problems. In this subsection, we present results for two image restoration problems based on images referred to as *Resolution* and *Cameraman* (see Figures 2 and 3). The images are 256 × 256 grayscale images; that is, $n = 256^2 = 65536$. The images are blurred by convolution with an 8 × 8 blurring mask and normally distributed noise with standard deviation 0.0055 added to the final signal (see problem 701 in [25]). The image restoration problem has the form (1.2), where $\tau = 0.00005$ and $\mathbf{A} = \mathbf{HW}$ is the composition of the 9×9 uniform blur matrix and the Haar discrete wavelet transform (DWT) operator. For these test problems, the continuation approach is no faster, and in some cases significantly slower, than the implementation without continuation. Therefore, we solved these test problems without the continuation technique. The results in Table 2 again indicate that the adaptive scheme yields much better performance as the error tolerance decreases.

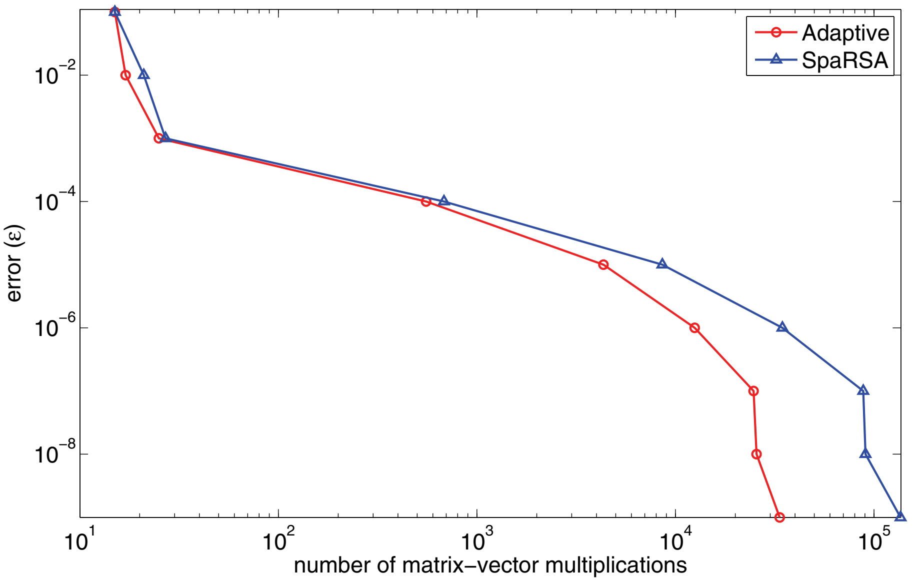

Figure 1. *Number of matrix-vector multiplications versus error.*

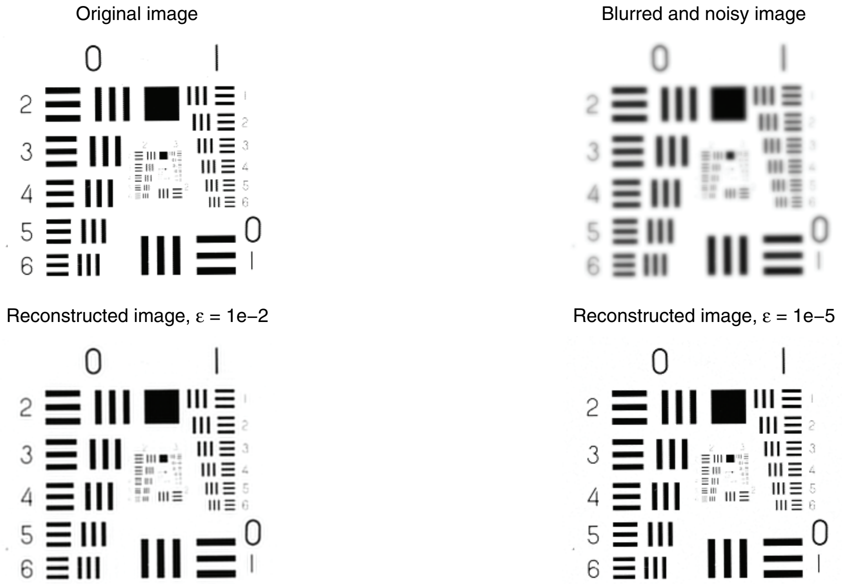

Figure 2. *Deblurring the Resolution image. The displayed reconstructions were obtained using Adaptive SpaRSA.*

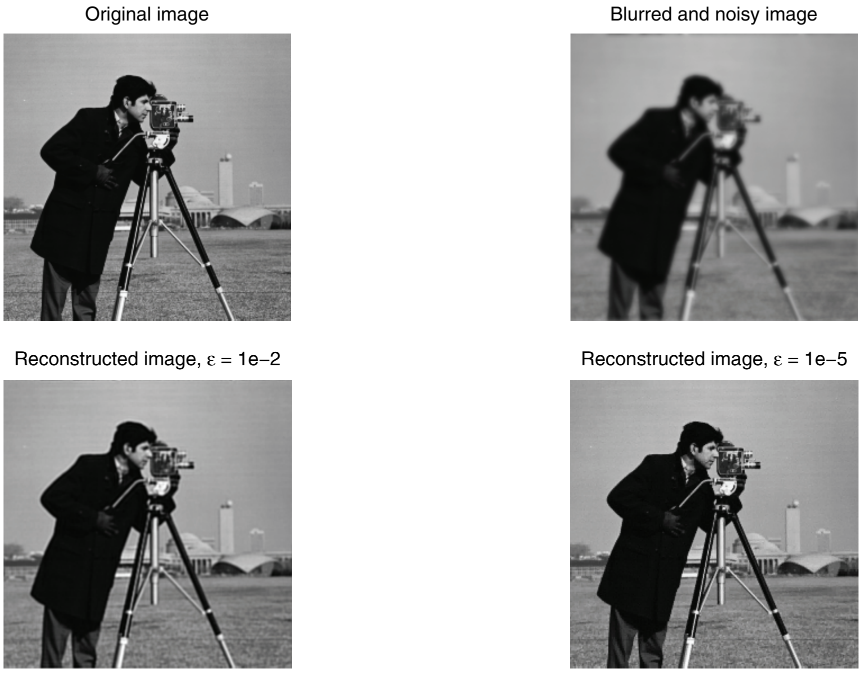

Figure 3. *Deblurring the Cameraman image. The displayed reconstructions were obtained using Adaptive SpaRSA.*

Table 2

Deblurring images.

ϵ			1e-2			1e-3			1e-4			1e-5
	Ax	cpu	Obj	Ax	cpu	Obj	Ax	cpu	Obj	Ax	cpu	Obj
Resolution												
SpaRSA	49	2.59	.4843	88	4.78	.3525	458	24.55	.2992	1631	87.49	.2970
SpaRSA+	37	1.93	.5619	73	4.03	.3790	422	22.04	.2988	1376	62.34	.2970
Adaptive	37	1.94	.5619	73	4.02	.3790	316	17.21	.2981	682	34.61	.2970
Cameraman												
SpaRSA	34	1.67	.3491	77	3.94	.2181	332	16.78	.1880	1374	69.98	.1868
SpaRSA+	35	1.73	.3380	63	3.28	.2232	295	14.33	.1880	994	52.36	.1868
Adaptive	35	1.72	.3380	63	3.29	.2232	215	11.07	.1880	602	31.43	.1868

6.3. Group-separable regularizer.

In this subsection, we examine performance using the group separable regularizers [28] for which

$$\psi(\mathbf{x}) = \tau \sum_{i=1}^{n} \|\mathbf{x}_{[i]}\|_2,$$

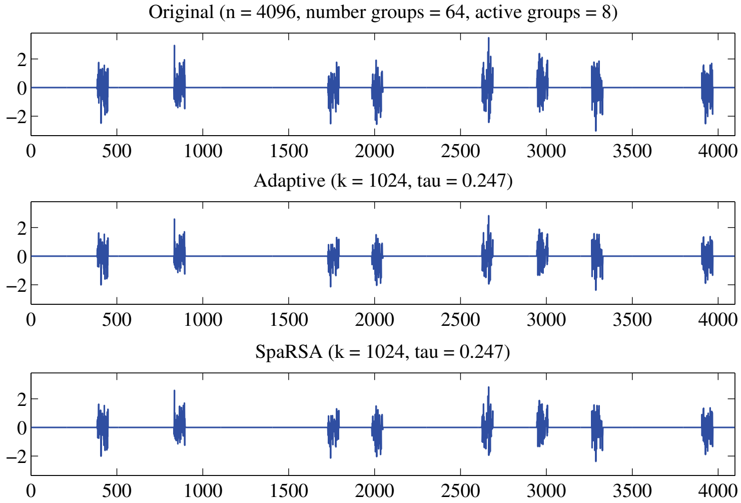

Figure 4. *Group-separable reconstruction.*

where $\mathbf{x}_{[1]}, \mathbf{x}_{[2]}, \ldots, \mathbf{x}_{[m]}$ are m disjoint subvectors of $\mathbf{x}$. The smooth part of ϕ can be expressed as $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$, where $\mathbf{A} \in \mathbb{R}^{1024\times 4096}$ was obtained by orthonormalizing the rows of a matrix constructed in subsection 6.1. The true vector $\mathbf{x}_{true}$ has 4096 components divided into $m = 64$ groups of length $l_i = 64$. $\mathbf{x}_{true}$ is generated by randomly choosing 8 groups and filling them with numbers chosen from a Gaussian distribution with zero mean and unit variance, while all other groups are filled with zeros. The target vector is $\mathbf{b} = \mathbf{A}\mathbf{x}_{true} + \mathbf{n}$, where $\mathbf{n}$ is Gaussian noise with mean zero and variance 10^{-4}. The regularization parameter is chosen as suggested in [28]: $\tau = 0.3\|\mathbf{A}^{\mathsf{T}}\mathbf{b}\|_\infty$. We ran 10 test problems with error tolerance $\epsilon = 10^{-5}$ and computed the average results. Adaptive SpaRSA solved the test problem in 0.8420 seconds with 67.4 matrix-vector multiplications, while the SpaRSA obtained similar performance: 0.8783 seconds and 69.1 matrix-vector multiplications. Figure 4 shows the result obtained by both methods for one sample.

6.4. Total variation phantom reconstruction. In this experiment, the image is the Shepp–Logan phantom of size 256 × 256 (see [5, 8]). The objective function was

$$\phi(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}(\mathbf{x}) - \mathbf{b}\|^2 + .01\mathrm{TV}(\mathbf{x}),$$

where $\mathbf{A}$ is a 6136×256^2 matrix corresponding to the evaluation of the two-dimensional Fourier transformation at 6136 locations in the plane (`masked_FFT` in the MATLAB software connected with [5]). The total variation (TV) regularization is defined as follows:

$$\mathrm{TV}(\mathbf{x}) = \sum_i \sqrt{\left(\triangle_i^h \mathbf{x}\right)^2 + \left(\triangle_i^v \mathbf{x}\right)^2},$$

where $\triangle_i^h$ and $\triangle_i^v$ are linear operators corresponding to horizontal and vertical first-order differences (see [6]). As seen in Table 3, Adaptive SpaRSA was faster than the original SpaRSA

Table 3
TV phantom reconstruction.

ϵ	1e-2			1e-3			1e-4		
	Ax	cpu	Obj	Ax	cpu	Obj	Ax	cpu	Obj
SpaRSA	14	2.55	36.7311	143	30.06	14.7457	2877	938.25	14.1433
Adaptive	14	2.57	36.7311	136	27.32	14.6840	731	185.62	14.1730

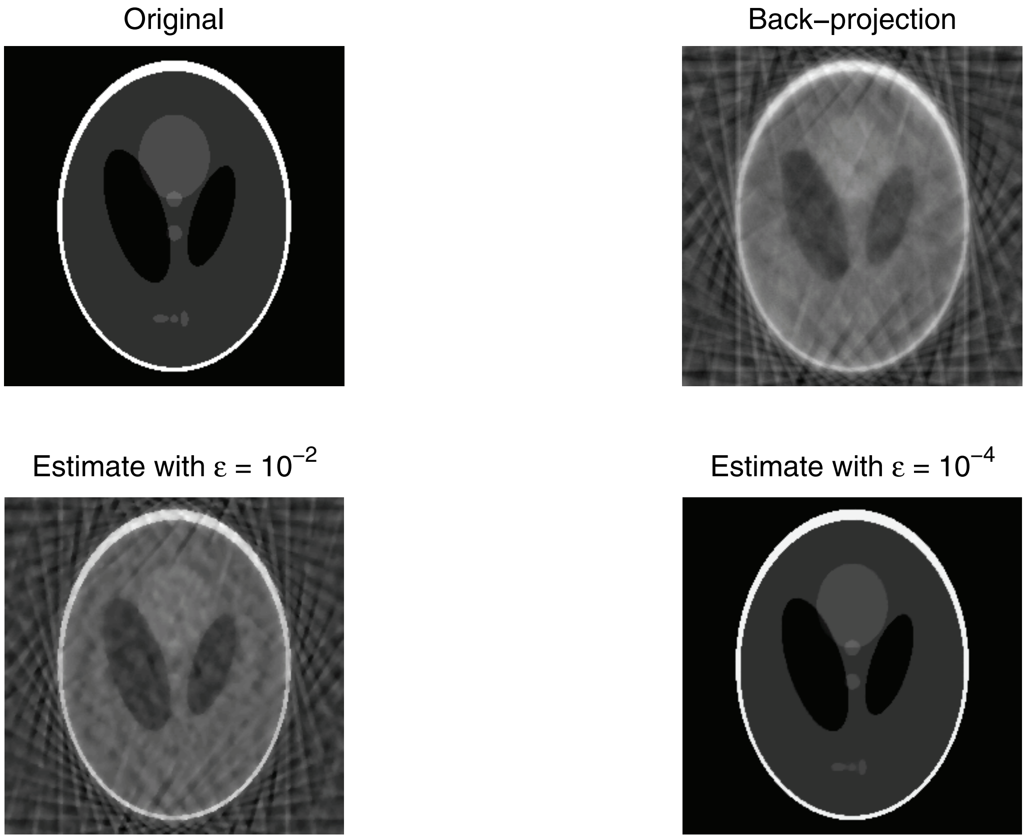

Figure 5. *Phantom reconstruction. The displayed reconstructions were obtained using Adaptive SpaRSA.*

when the error tolerance was sufficiently small. The reconstructed images corresponding to $\epsilon = 10^{-2}$ and $\epsilon = 10^{-4}$ are shown in Figure 5.

6.5. Comparisons with alternating direction method of multipliers.

In this subsection, we compare the execution times between Adaptive SpaRSA, SpaRSA, and the alternating direction method of multipliers SALSA [1]. We first run SpaRSA and stop when

$$\alpha_k \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_\infty \le 10^{-5}.$$

The objective function value obtained by SpaRSA is used as the stopping criterion for the other methods. That is, the algorithms are terminated when their objective function values reach SpaRSA's value.

We consider four test problems. Two of the problems are identical to the ones previously presented: the ℓ_2–ℓ_1 problems of section 6.1 and the group-separable regularizer problems of section 6.3. The two remaining problems were taken from [1] since the critical inversion step appearing in SALSA was worked out for these specific problems. The first problem, a Cameraman deblurring problem with wavelets, is Experiment 1 in section IV.A of [1]. The operator in the image reconstruction problem has the form $\mathbf{A} = \mathbf{HW}$, where $\mathbf{H}$ is a 9×9 uniform blur matrix, $\mathbf{W}$ is a redundant Haar wavelet frame with four levels, and the noise variance is 0.56. The second problem, the TV reconstruction of the 128×128 Shepp–Logan phantom, was introduced in section IV.C of [1]. For these two image reconstruction problems, efficient methods have been developed in SALSA for solving the linear system

$$(\mathbf{A}^\mathsf{T}\mathbf{A} + \mu\mathbf{I})\mathbf{x}_{k+1} = \mathbf{A}^\mathsf{T}\mathbf{b} + \mu\mathbf{x}'_k$$

which arises in each iteration. In Table 4, we report not only the CPU times but also the final mean squared error (mse) or signal-to-noise ratio improvement (isnr) in dB.

Table 4
Comparisons of SpaRSA, Adaptive, and SALSA. CPU time in seconds.

			SpaRSA	Adaptive	SALSA
$\ell_2 - \ell_1$	$n = 1024$	cpu	2.63	1.46	4.89
	$k = 256$	mse	3.85e-7	3.84e-7	3.74e-7
	$n = 4096$	cpu	19.90	11.26	93.94
	$k = 1024$	mse	5.74e-7	5.91e-7	5.36e-7
Deblurring	Cameraman	cpu	309.15	146.81	24.50
		isnr	7.46	7.46	7.44
Group		cpu	1.13	0.91	14.09
		mse	6.86e-3	6.86e-3	6.86e-3
Phantom	$\tau = 9e - 5$	cpu	34.95	31.23	16.73
		mse	9.18e-6	7.72e-6	2.73e-7
	$\tau = 1e - 3$	cpu	7.64	7.82	33.89
		mse	7.69e-6	7.83e-6	3.85e-6

The performance results in Table 4 indicate that the Adaptive algorithm had the best performance for the ℓ_2–ℓ_1 problems and the group-separable regularizer problems. For the Cameraman deblurring problem, SALSA was the fastest. For the Shepp–Logan phantom, SALSA was the fastest when the regularization was $9 \times 10^{-5}\mathrm{TV}(\mathbf{x})$ (suggested in [1]) while SpaRSA was the fastest when the regularization was $1 \times 10^{-3}\mathrm{TV}(\mathbf{x})$.

7. Conclusions. The convergence properties of the sparse reconstruction by separable approximation (SpaRSA) algorithm of Wright, Nowak, and Figueiredo [28] are analyzed. We establish sublinear convergence when ϕ is convex, and the GLL reference function value [17] is employed. When ϕ is strongly convex, the convergence is R-linear. For a reference function value which satisfies (R1)–(R3), we prove the existence of a convergent subsequence of iterates that approaches a stationary point. For a slightly stronger version of (R3), given in (5.2), we show that sublinear or linear convergence again hold when ϕ is convex or strongly convex, respectively. In a series of numerical experiments, it is shown that Adaptive SpaRSA, based on a relaxed choice of the reference function value and a cyclic BB iteration [12, 18], often yields much faster convergence than the original SpaRSA, especially when the error tolerance is small. For problems with special structure, the alternating direction method of multipliers SALSA could give very good performance relative to the other methods.

Acknowledgments. Constructive comments by the reviewers are gratefully acknowledged. In particular, their comments led to a sharpening of the convergence theory, improvements in the exposition, and additional insights into the relative performance of the methods.

REFERENCES

[1] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, *Fast image recovery using variable splitting and constrained optimization*, IEEE Trans. Image Process., 19 (2010), pp. 2345–2356.

[2] J. Barzilai and J. M. Borwein, *Two point step size gradient methods*, IMA J. Numer. Anal., 8 (1988), pp. 141–148.

[3] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[4] S. Becker, J. Bobin, and E. J. Candès, *NESTA: A fast and accurate first-order method for sparse recovery*, SIAM J. Imaging Sci., 4 (2011), pp. 1–39.

[5] J. Bioucas-Dias and M. Figueiredo, *TwIST: Two-Step Iterative Shrinkage/Thresholding Algorithm for Linear Inverse Problems*, http://www.lx.it.pt/~bioucas/TwIST/TwIST.htm.

[6] J. Bioucas-Dias, M. Figueiredo, and J. P. Oliveira, *Total variation-based image deconvolution: A majorization-minimization approach*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 2, 2006, pp. 861–864.

[7] R. Broughton, I. Coope, P. Renaud, and R. Tappenden, *A Barzilai-Borwein ℓ_1-Regularized Least Squares Algorithm for Compressed Sensing*, Technical report, 2009; available online from http://arxiv.org/abs/0907.3340.

[8] E. J. Candès and J. Romberg, *Practical signal recovery from random projections*, in Proceedings of the SPIE Wavelet Applications in Signal and Image Processing Conference XI, Vol. 5914, 2005.

[9] 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 Trans. Image Process., 7 (1998), pp. 319–335.

[10] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comput., 20 (1998), pp. 33–61.

[11] Y. H. Dai, *Alternate stepsize gradient method*, Optimization, 52 (2003), pp. 395–415.

[12] Y. H. Dai, W. W. Hager, K. Schittkowski, and H. Zhang, *The cyclic Barzilai-Borwein method for unconstrained optimization*, IMA J. Numer. Anal., 26 (2006), pp. 604–627.

[13] I. Daubechies, M. Defrise, and C. D. Mol, *An iterative thresholding algorithm for linear problems with a sparsity constraint*, Comm. Pure Appl. Math., 57 (2004), pp. 1413–1457.

[14] 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 J. Sel. Top. Signal Process., 1 (2007), pp. 586–597.

[15] T. Figueiredo and R. D. Nowak, *An EM algorithm for wavelet-based image restoration*, IEEE Trans. Image Process., 12 (2003), pp. 906–916.

[16] A. Friedlander, J. M. Martínez, B. Molina, and M. Raydan, *Gradient method with retards and generalizations*, SIAM J. Numer. Anal., 36 (1999), pp. 275–289.

[17] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal., 23 (1986), pp. 707–716.

[18] W. W. Hager and H. Zhang, *A new active set algorithm for box constrained optimization*, SIAM J. Optim., 17 (2006), pp. 526–557.

[19] E. T. Hale, W. Yin, and Y. Zhang, *Fixed-point continuation for ℓ_1-minimization: Methodology and convergence*, SIAM J. Optim., 19 (2008), pp. 1107–1130.

[20] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, *An interior-point method for large-scale ℓ_1-regularized least squares*, IEEE J. Sel. Top. Signal Process., 1 (2007), pp. 606–617.

[21] Z. Lu and Y. Zhang, *An Augmented Lagrangian Approach for Sparse Principal Component Analysis*, Technical report, Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada, 2009.

[22] Y. Nesterov, *Gradient Methods for Minimizing Composite Objective Function*, CORE Discussion Papers 2007/76, Center for Operations Research and Econometrics (CORE), Université catholique de Louvain, Louvain-la-Neuve, Belgium, 2007.

[23] M. Raydan and B. F. Svaiter, *Relaxed steepest descent and Cauchy-Barzilai-Borwein method*, Comput. Optim. Appl., 21 (2002), pp. 155–167.

[24] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[25] E. van den Berg, M. P. Friedlander, G. Hennenfent, F. J. Herrmann, R. Saab, and O. Yilmaz, *Algorithm* 890: *Sparco: A testing framework for sparse reconstruction*, ACM Trans. Math. Software, 35 (2009), pp. 1–16.

[26] L. Vandenberghe, *Gradient Methods for Nonsmooth Problems*, http://www.ee.ucla.edu/~vandenbe/ee236c.html.

[27] C. Vonesch and M. Unser, *Fast iterative thresholding algorithm for wavelet-regularized deconvolution*, in Proceedings of the SPIE Optics and Photonics 2007 Conference on Mathematical Methods: Wavelet XII, Vol. 6701, San Diego, CA, 2007, pp. 1–5.

[28] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, *Sparse reconstruction by separable approximation*, IEEE Trans. Signal Process., 57 (2009), pp. 2479–2493.