AN EFFICIENT PROJECTION METHOD FOR NONLINEAR INVERSE PROBLEMS WITH SPARSITY CONSTRAINTS

DEREN HAN*, ZEHUI JIA AND YONGZHONG SONG

School of Mathematical Sciences
Key Laboratory for NSLSCS of Jiangsu Province
Nanjing Normal University
Nanjing 210023, China

DAVID Z. W. WANG

School of Civil and Environmental Engineering
Nanyang Technological University
50 Nanyang Avenue, 639798 Singapore, Singapore

(Communicated by Yunmei Chen)

Abstract. In this paper, we propose a modification of the accelerated projective steepest descent method for solving nonlinear inverse problems with an \( \ell_1 \) constraint on the variable, which was recently proposed by Teschke and Borries (2010 Inverse Problems 26 025007). In their method, there are some parameters need to be estimated, which is a difficult task for many applications. We overcome this difficulty by introducing a self-adaptive strategy in choosing the parameters. Theoretically, the convergence of their algorithm was guaranteed under the assumption that the underlying mapping \( F \) is twice Fréchet differentiable together with some other conditions, while we can prove weak and strong convergence of the proposed algorithm under the condition that \( F \) is Fréchet differentiable, which is a relatively weak condition. We also report some preliminary computational results and compare our algorithm with that of Teschke and Borries, which indicate that our method is efficient.

1. Introduction. We consider the computation of an approximate solution of the nonlinear inverse problem

\[
F(f) = y,
\]

where \( F : X \to Y \) is a possibly ill-posed operator between Hilbert spaces \( X \) and \( Y \). Since it is often the case that only noise data \( y^\delta \) with \( \|y^\delta - y\| \leq \delta \) are available, problem (1) has to be stabilized by regularization methods. A usually used regularization technique is to add a \( \|x\|_p \) term to the least squares of the equation (1), where \( 0 \leq p < 2 \). In many applications, the solution \( f \) of the equation (1) has a sparse series expansion \( f = \sum_{\lambda \in \Lambda} x_\lambda \phi_\lambda \) with respect to a preassigned basis or frame \( \{\phi_\lambda\}_{\lambda \in \Lambda} \subset X \), i.e., the series expansion of \( f \) has only a very small number of nonzero coefficients \( x_\lambda \). This property can be characterized by choosing \( p = 0 \), leading to a NP-hard problem. Since the \( \ell_1 \)-norm is the best convex approximation...
of the zero-norm in the unit $\ell_1$ ball, $p = 1$ is currently the most popular choice. In this paper, we focus our attention on the sparsity measure $\| \cdot \|_1$. That is, we consider the regularized problem

\[
(2) \quad \min_x \left\| F \left( \sum_{\lambda \in \Lambda} x_\lambda \phi_\lambda \right) - y \right\|^2 + 2\alpha \| x \|_1,
\]

where $\alpha > 0$ is the parameter balancing the data-fidelity term and the $\ell_1$ term.

For linear inverse problems, there are recently several approximation/regularization methods with sparsity constraints, e.g., in [2, 3, 4, 6, 7, 11, 13, 20]. Generalize the classical Landweber iteration for linear inverse problems to solve (2), resulting in the following recursion

\[
(3) \quad x^{n+1}_\lambda = S_\alpha \left( x^n_\lambda + \gamma \left( F^* \left( y - F \left( \sum_\eta x^n_\eta \phi_\eta \right) \right), \phi_\lambda \right) \right),
\]

where $\gamma > 0$ is a selected parameter and $S_\alpha$ is the soft-shrinkage operator, which is defined as that for a scalar $\xi$

\[
S_\alpha(\xi) := \xi - \min\{\alpha, |\xi|\} \frac{\xi}{|\xi|},
\]

where $\xi = 0$ if $\xi = 0$; for a vector $x$, $S_\alpha(x)$ is also a vector whose $i$-th element is $S_\alpha(x_i)$. This procedure, although is quite simple, is known to converge usually slowly. A direct generalization of (3) to the nonlinear inverse problems yields the iteration scheme

\[
(4) \quad x^{n+1}_\lambda = S_\alpha \left( x^n_\lambda + \gamma \left( F^* \left( y - F \left( \sum_\eta x^n_\eta \phi_\eta \right) \right), \phi_\lambda \right) \right),
\]

which also converges slowly, as that observed by Ramlau and Teschke in [19].

One of the key factors that leads to the slow convergence of (3) for linear inverse problems is that, after an initially relatively fast convergence, the iterate traps in the overshooting of the $\ell_1$ penalty and it takes very long to re-correct back. This observation inspired the authors in [5] to propose the intuitive way of forcing the iterates remaining within a particular $\ell_1$ ball. That is, instead of solving the model (2), they propose to solve the ‘constrained’ one

\[
(5) \quad \min_{x \in B_R} \left\| F \left( \sum_{\lambda \in \Lambda} x_\lambda \phi_\lambda \right) - y \right\|^2,
\]

where $B_R$ is the $\ell_1$ ball

\[
(6) \quad B_R := \{ x \in \ell_2 \mid \| x \|_1 \leq R \}.
\]

Iteration scheme (3) is correspondingly modified by replacing the soft-shrinkage operator $S_\alpha$ with the operator $P_R$, which is the projection operator on $B_R$, resulting in the projected Landweber iteration

\[
(7) \quad x^{n+1}_\lambda = P_R \left( x^n_\lambda + \gamma \left( F^* \left( y - F \left( \sum_\eta x^n_\eta \phi_\eta \right) \right), \phi_\lambda \right) \right).
\]

For a closed convex set $\mathcal{C}$ and any $x \in X$, the projection of $x$ onto $\mathcal{C}$ is defined as the unique point in $\mathcal{C}$ for which the $\ell_2$ distance to $x$ is minimal.
Recently, by generalizing the iteration scheme (7) to the nonlinear inverse problem, Teschke and Borries [21] proposed a new iteration procedure and managed to prove the convergence of their algorithm under the assumption that the operator $F$ is twice Fréchet differentiable and the following three requirements hold true:

\begin{align}
(8) \quad x^n \rightarrow x^* \Rightarrow F\left( \sum_{\lambda} x^n_{\lambda} \phi_{\lambda} \right) \rightarrow F\left( \sum_{\lambda} x^*_{\lambda} \phi_{\lambda} \right), \\
(9) \quad F'\left( \sum_{\lambda} x^n_{\lambda} \phi_{\lambda} \right)^* y \rightarrow F'\left( \sum_{\lambda} x^*_{\lambda} \phi_{\lambda} \right)^* y \quad \text{for all } y, \\
(10) \quad \left\| F'\left( \sum_{\lambda} x_{\lambda} \phi_{\lambda} \right) - F'\left( \sum_{\lambda} x'_{\lambda} \phi_{\lambda} \right) \right\| \leq L \| x - x' \|.
\end{align}

Some numerical results were also presented to illustrate the algorithm. Here and throughout the paper, $D(x)$ is the least squares of the equation (1)

\begin{equation}
(11) \quad D(x) := \| F(x) - y \|^2.
\end{equation}

The detail of Teschke and Borries’s algorithm is as follows [21].

**Algorithm 1.1** Projected steepest descent method for nonlinear inverse problems.

1: Initialization. Choose some initial guess $x^0$ and $R$, and set

\begin{equation}
(12) \quad r = \max \left\{ 2 \sup_{x \in B_R} \| F'(x) \|^2, 2L\sqrt{D(x^0)} \right\}.
\end{equation}

Choose a parameter $0 < q < 1$.

2: while a preassigned stopping criterion is not satisfied and the maximum number of iterations is not reached do

3: \quad $\beta^n = \max \left\{ \sup_{x \in B_R} \frac{\| F'(x) \|^2}{L\sqrt{D(x)}} \right\}$.

4: \quad $x^{n+1} = P_R \left( x^n + \frac{\beta^n}{r} F'(x^{n+1})^* (y - F(x^n)) \right)$: by fixed point iteration.

5: \quad Verify (B2): $\beta^n \| F(x^{n+1}) - F(x^n) \|^2 \leq \frac{\delta}{2} \| x^{n+1} - x^n \|^2$.

\quad If (B2) is satisfied increase $n$ and go to 3; otherwise set $\beta^n = q \cdot \beta^n$ and go to 4.

6: end while

Checking their algorithm, we find that the parameter $\beta^n$ plays the role of step size during the iterations, which is crucial to the efficiency the algorithm. From the numerical point of view, a larger of such a step size is more desirable. Thus, a larger of the right hand of (B2) in Algorithm 1.1 is needed. In this paper, we relax it by a factor $2(1 - \delta)$ with $0 < \delta < 1$ and being close to 0, i.e., we replace (B2) by the following criterion

\begin{equation}
(13) \quad \beta^n \| F(x^{n+1}) - F(x^n) \|^2 \leq (1 - \delta) r \| x^{n+1} - x^n \|^2.
\end{equation}

More importantly, due to the trial point $x^{n+1}$ is obtained in an implicit way, a suitable choice of the initial guess of $\beta^n$ is needed. The guess in the step 3 may be too large, at least for most steps. In this paper, we choose the initial guess of the $\beta^n$ as the one of the last iteration. To avoid too conservative of this choice, we allow to increase it when necessary. Such a ‘self-adaptive’ choice of the step size
was adopted in [25] for solving the multiple-sets split feasibility problems, where the numerical results show that such a simple strategy can introduce much efficiency of the algorithm.

Throughout our analysis, we just assume that the operator $F$ is continuously differentiable, but not necessarily twice differentiable as that assumed in [21]. This weaker condition on guaranteeing convergence can enlarge the range of applications of the algorithm, which makes the algorithm more practical.

The rest of the paper is organized as follows. In the next section, we summarize some necessary preliminary results. In Section 3, we describe the algorithm formally and prove its weak and strong convergence in Section 3.1 and Section 3.2, respectively. We report some preliminary numerical results and compare the new algorithm with that of Teschke and Borries [21] in Section 4. Finally, we conclude the paper with Section 5.

2. Preliminaries. Throughout the paper, we adopt the standard notation as that in [21]. For a countable index set $\Lambda$, we denote by $\ell_p(\Lambda)$ the space of sequences $x$ satisfying $\|x\|_p < \infty$. For a given Hilbert space $X$, let $\{\phi_\lambda\}_{\lambda \in \Lambda} \subset X$ be a preassigned frame. Then there exist constants $0 < A \leq B < \infty$,

\begin{equation}
A\|f\|_X^2 \leq \sum_{\lambda \in \Lambda} |\langle f, \phi_\lambda \rangle|^2 \leq B\|f\|_X^2, \quad \text{for all } f \in X.
\end{equation}

This frame allows us to consider the so-called frame operator $S : X \to \ell_2(\Lambda)$ that is defined by $Sf = \{\langle f, \phi_\lambda \rangle\}_{\lambda \in \Lambda}$. Consequently, the computation of a solution $f$ is translated into finding a corresponding sequence $x \in \ell_2(\Lambda)$. Let $\hat{F} := F \circ S^* : \ell_2(\Lambda) \to Y$. Then, problem (5) can be formulated as

\[ \min_{x \in B_R} \|\hat{F}(x) - y\|^2. \]

In the rest of the paper, we still use $F$ to denote $\hat{F}$ which should not introduce ambiguity and focus our attention on the model

\[ \min_{x \in B_R} \|F(x) - y\|^2. \]

Since we consider the constrained model (15), projection operator is often used. For a nonempty closed convex subset $C$ of $X$ and any vector $x \in X$,

\[ P_C(x) := \arg\min_{z \in C} \|x - z\|. \]

We now summarize some basic properties of the projection operator.

**Lemma 2.1.** Let $C$ be a nonempty closed convex subset of $X$, for any $u, v \in X$ and any $w \in C$, the following properties hold.

1. $u \in C \iff P_C(u) = u$.
2. $\langle u - P_C(u), w - P_C(u) \rangle \leq 0$.
3. $\|P_C(u) - P_C(v)\| \leq \|u - v\|$.

**Proof.** See Facchinei and Pang [10].

For the model (15), we have the following result, which provides a necessary condition for a point to be its minimizer.
Lemma 2.2. Suppose that $F$ is continuously Fréchet differentiable. If the vector $\tilde{x}_R \in \ell_2(\Lambda)$ is a minimizer of $D(x)$ on $B_R$, then for any $\gamma > 0$,

$$P_R(\tilde{x}_R + \gamma F'(\tilde{x}_R)^*(y - F(\tilde{x}_R))) = \tilde{x}_R,$$

which is equivalent to

$$\langle F'(\tilde{x}_R)^*(y - F(\tilde{x}_R)), w - \tilde{x}_R \rangle \leq 0, \quad \text{for all } w \in B_R.$$

Proof. Since $F$ is Fréchet differentiable, $D$ is differentiable with

$$\nabla D(x) = -2F'(x)^*(y - F(x)).$$

Since $B_R$ is convex, for any $x \in B_R$, the feasible direction set at $x$ is [1]

$$\mathcal{F}(x) = \{s \mid s = x' - x, \ x' \in B_R\}.$$ Because $\tilde{x}_R$ is a minimizer of $D$, any feasible direction should not be a descent direction of $D$, i.e.,

$$\langle F'(\tilde{x}_R)^*(y - F(\tilde{x}_R)), w - \tilde{x}_R \rangle \leq 0, \quad \text{for all } w \in B_R.$$ Particularly, since $P_R(\tilde{x}_R + \gamma F'(\tilde{x}_R)^*(y - F(\tilde{x}_R))) \in B_R$,

$$(16) \quad \langle F'(\tilde{x}_R)^*(y - F(\tilde{x}_R)), P_R(\tilde{x}_R + \gamma F'(\tilde{x}_R)^*(y - F(\tilde{x}_R))) - \tilde{x}_R \rangle \leq 0.$$ On the other hand, setting $u := \tilde{x}_R + \gamma F'(\tilde{x}_R)^*(y - F(\tilde{x}_R))$ and $w := \tilde{x}_R$ in the second inequality of Lemma 2.1 yields

$$(17) \quad \tilde{x}_R - P_R(\tilde{x}_R + \gamma F'(\tilde{x}_R)^*(y - F(\tilde{x}_R))) \leq 0.$$ Multiplying $\gamma$ to (16) and adding the resulted inequality with (17) we get

$$\langle \tilde{x}_R - P_R(\tilde{x}_R + \gamma F'(\tilde{x}_R)^*(y - F(\tilde{x}_R))), \tilde{x}_R - P_R(\tilde{x}_R + \gamma F'(\tilde{x}_R)^*(y - F(\tilde{x}_R))) \rangle \leq 0,$$

which implies

$$\tilde{x}_R = P_R(\tilde{x}_R + \gamma F'(\tilde{x}_R)^*(y - F(\tilde{x}_R))).$$ This completes the proof. \hfill \Box

Let

$$(18) \quad e(x, \gamma) := x - P_R(x + \gamma F'(x)^*(y - F(x)))$$

denote the residual error, then finding a minimizer of (15) is amount to finding a zero point of $e(x, \gamma)$. In addition, $\|e(x, \gamma)\|$ can be viewed as an error bound of measuring how much $x$ fails to be a minimizer of (15), and serves as a stopping criterion of numerical methods.

For given $x \in \mathcal{C}$, the magnitude $\|e(x, \gamma)\|$ has the following property, which will be used in the following sections.

Lemma 2.3. Suppose $x \in B_R$ is not a minimizer of (15) and $\tilde{\gamma} \geq \gamma > 0$, then we have

$$\|e(x, \gamma)\| \geq \|e(x, \tilde{\gamma})\|$$

and

$$\frac{\|e(x, \tilde{\gamma})\|}{\tilde{\gamma}} \leq \frac{\|e(x, \gamma)\|}{\gamma}.$$
Proof. The proof skill is similar to that in [26], which is for the study of the monotone variational inequality problem. Setting \( w := P_R(x + \hat{\gamma}F'(x)^*(y - F(x))) \) and \( u := x + \gamma F'(x)^*(y - F(x)) \) in the second inequality of Lemma 2.1, and using the relation
\[
P_R(x + \gamma F'(x)^*(y - F(x))) - P_R(x + \hat{\gamma}F'(x)^*(y - F(x))) = e(x, \hat{\gamma}) - e(x, \gamma),
\]
we have
\[
(e(x, \gamma) + \gamma F'(x)^*(y - F(x)), e(x, \hat{\gamma}) - e(x, \gamma)) \geq 0.
\]
In a similar way, we can get
\[
(e(x, \hat{\gamma}) + \hat{\gamma}F'(x)^*(y - F(x)), e(x, \gamma) - e(x, \hat{\gamma})) \geq 0.
\]
Multiplying the first inequality by \( \hat{\gamma} \) and the second one by \( \gamma \), and then adding the resulting inequalities yields
\[
\langle \hat{\gamma}e(x, \gamma) - \gamma e(x, \hat{\gamma}), e(x, \hat{\gamma}) - e(x, \gamma) \rangle \geq 0,
\]
i.e.,
\[
\gamma \|e(x, \hat{\gamma})\|^2 + \hat{\gamma} \|e(x, \gamma)\|^2 \leq (\gamma + \hat{\gamma}) \langle e(x, \gamma), e(x, \hat{\gamma}) \rangle \leq (\gamma + \hat{\gamma}) \|e(x, \hat{\gamma})\| \|e(x, \gamma)\|,
\]
where the second inequality follows from the Cauchy-Schwarz inequality. Rearranging terms, we have
\[
\langle (\gamma \|e(x, \hat{\gamma})\| - \hat{\gamma} \|e(x, \gamma)\|) (\|e(x, \hat{\gamma})\| - \|e(x, \gamma)\|) \rangle \leq 0,
\]
and the assertion follows immediately.

The following assertion was proved in [21] under the conditions that \( F \) is twice Fréchet differentiable, and \( \beta L \sqrt{D(x)} \leq r/2 \). In the following, we prove a similar result under the weaker conditions that \( F \) is continuously Fréchet differentiable, and \( \beta L \sqrt{D(x)} < r \).

**Lemma 2.4.** Assume that \( F \) is continuously Fréchet differentiable and \( \beta > 0 \). For any fixed \( x \in B_R \) assume
\[
\beta L \sqrt{D(x)} < r,
\]
and define the functional \( \Phi_\beta(\cdot, x) \) by
\[
\Phi_\beta(w, x) := \|F(w) - y\|^2 - \|F(w) - F(x)\|^2 + \frac{r}{\beta} \|w - x\|^2.
\]
Then there exists a unique \( w \in B_R \) that minimizes the restriction to \( B_R \) of \( \Phi_\beta(w, x) \). We denote this minimizer by \( \hat{w} \) which is given by
\[
\hat{w} = P_R \left( x + \frac{\beta}{r} F'(\hat{w})^*(y - F(x)) \right).
\]

Proof. For simplicity of notation, we define
\[
J(w) := \Phi_\beta(w, x) = \|F(w) - y\|^2 - \|F(w) - F(x)\|^2 + \frac{r}{\beta} \|w - x\|^2.
\]
Since \( F \) is Fréchet differentiable, \( J(w) \) is continuously differentiable with
\[
\nabla J(w) = 2 \left( F'(w)^*(F(x) - y) + \frac{r}{\beta} (w - x) \right).
\]
For any \( w, w' \in B_R \), \( w \neq w' \), we have
\[
\langle \nabla J(w) - \nabla J(w'), w - w' \rangle \\
= \frac{2r}{\beta} \left( \| w - w' \|^2 - \frac{\beta}{r} \| w - w' \| \| F'(w) - F'(w') \| \| F(x) - y \| \right) \\
\geq \frac{2r}{\beta} \left( \| w - w' \|^2 - \frac{\beta}{r} \| w - w' \| (L\| w - w' \|) \left( \frac{r}{\beta L} \right) \right) \\
= 0,
\]
which means that \( \nabla J \) is strictly monotone. Here the first inequality follows from the Cauchy-Schwarz inequality, the second one from the Lipschitz continuity of \( F' \) and the assumption (22). Consequently, \( J \) is strictly convex. Since \( B_R \) is closed, convex, and \( J(w) \) is strictly convex, it has a unique minimizer on \( B_R \). From the first-order optimality condition, this minimizer \( \hat{w} \) can be characterized as
\[
\hat{w} = P_R(\hat{w} - \gamma \nabla J(\hat{w})),
\]
where \( \gamma \) is an arbitrary (but fixed) positive number. Setting \( \gamma := \beta/(2r) \), we obtain the desired assertion immediately. \( \square \)

The unique minimizer of \( \Phi_{\beta} \) is given in an implicit way, in the sense that both sides of the equality (24) involve \( \hat{w} \). To find an approximation of the minimizer, a simple way, as suggested in [21] is to adopt a fixed-point iteration
\[
\hat{x}^{l+1} = \Psi(\hat{x}^l) := P_R(x + \beta/r F'((\hat{x}^l)^*)(y - F(x))).
\]
To ensure the convergence of such a fixed-point iteration, it is sufficient that the mapping \( \Psi \) is contractive. The following lemma shows this.

**Lemma 2.5.** Assume that \( \beta L \sqrt{D(x)} \leq (1 - \delta) r \), where \( 0 < \delta < 1 \). The map \( \Psi \) defined in (26) is contractive and as a consequence, the iteration (26) is convergent.

**Proof.** For any \( w, w' \in B_R \), we have
\[
\| \Psi(w) - \Psi(w') \| \\
= \| P_R(x + \beta/r F'((w)^*)(y - F(x))) - P_R(x + \beta/r F'(w')^*)(y - F(x)) \| \\
\leq \frac{\beta}{r} \| (F'(w)^* - F'(w')^*)(y - F(x)) \| \\
\leq \frac{\beta}{r} L \sqrt{D(x)} \| w - w' \| \\
\leq (1 - \delta) \| w - w' \|.
\]
The proof is complete. \( \square \)

The fixed-point (26) may not an efficient scheme in finding \( \hat{w} \), since it is a projection method with a constant step size \( \beta/(2r) \). In applications, a variable step size selected in a self-adaptive manner is usually more favorable, i.e., instead of using (26), we suggest to use the iteration
\[
\hat{x}^{l+1} := P_R(x + \alpha_l \beta/r F'((\hat{x}^l)^*)(y - F(x))),
\]
where \( \alpha_{\text{max}} \geq \alpha_l \geq \alpha_{\text{min}} > 0 \) is a parameter which is selected in a self-adaptive manner as suggested in [15, 14], and the interested readers are referred to these two references for details.
3. The algorithm and its convergence. In this section, we first present the new algorithm for solving (15) and then analyze its convergence.

3.1. Algorithm description and properties. Following we describe the algorithm in detail.

Algorithm 3.1 A self-adaptive projection method for nonlinear inverse problems.

1: Given a nonnegative sequence $\tau_n$ with $\sum_{n=0}^{\infty} \tau_n < +\infty$, $\delta \in (0, 1)$, $\mu \in (0, 1)$, $\rho \in (0, 1)$, $\varepsilon > 0$, $\beta_0 > 0$, and an arbitrary initial point $x^0$. Set $\gamma_0 = \beta_0$, and $n = 0$.

2: Find the smallest nonnegative integer $l_n$ such that $\beta_{n+1} = \mu^{l_n} \gamma_n$ and use the self-adaptive iteration scheme (28) to get $x^{n+1}$ satisfying

$$x^{n+1} = P_R \left( x^n + \frac{\beta_{n+1}}{r} F'(x^{n+1})^* (y - F(x^n)) \right),$$

and

$$\beta_{n+1} \| F(x^{n+1}) - F(x^n) \|^2 \leq (1 - \delta) r \| x^{n+1} - x^n \|^2.$$

3: If

$$\beta_{n+1} \| F(x^{n+1}) - F(x^n) \|^2 \leq \rho r \| x^{n+1} - x^n \|^2,$$

then set $\gamma_{n+1} = (1 + \tau_{n+1}) \beta_{n+1}$; otherwise set $\gamma_{n+1} = \beta_{n+1}$.

4: If $\| e(x^{n+1}, \beta_{n+1}) \| \leq \varepsilon$, stop; otherwise set $n := n + 1$ and go to S2.

Remark 1. This new method is a modification of the accelerated projected steepest descent method proposed by Teschke and Borries [21]. Comparing these two methods, we can find that there are the following differences:

- We require the operator $F$ be continuously Fréchet differentiable, while Teschke and Borries [21] required $F$ to be twice Fréchet differentiable. The weaker requirement enables our algorithm to be applied in wider fields.
- The factor in our application criterion of $\beta_{n}$ is $1 - \delta$, while it is $1/2$ in [21]. This slight change makes it possible to accept a larger $\beta_{n}$, which may affect the algorithm’s efficiency.
- The manner in select the initial value of $\beta_{n}$ is different. In [21], the authors select $\beta_{n}$ to be

$$\max \left\{ \sup_{x \in B_R} \frac{\| F'(x) \|}{L \sqrt{\| D(x) \|}}, \sqrt{\frac{\| D(x) \|}{L \sqrt{\| D(x) \|}}} \right\},$$

which may be too large, leading to a large number of iterations to find a suitable step size. While in our algorithm, we just select the initial guess using the information from the last iteration.

- More importantly, we adopt a ‘self-adaptive’ procedure in select the next trial of $\beta_{n}$, i.e., Step S3. Such a self-adaptive strategy was first proposed by He et al. [15] for solving variational inequality problems with strongly monotone and Lipschitz continuous mappings. It was recently extended by Zhang et al. [25] to solve the multiple-sets split feasibility problem. The numerical results reported in these papers indicated that the strategy works very well.

- Both algorithms need to find a fixed point of (24) which is used as the next iteration $x^{n+1}$. In [21], the authors used a constant step size $\beta/(2r)$ while in our algorithm, we also adopt a variable step size selecting in a self-adaptive manner. This can enhance the performance of the algorithm greatly.
Therefore, we now pay our attention to the other two assertions. By this, we mean
\[ \text{Lemma 3.1.} \]
Thus, the sequence \( \{D^n\} \) is well-defined.

Rearranging terms, we obtain
\[ \{D^n\} \text{ is uniformly bounded from above. Hence, the whole algorithm is well-defined.} \]

We conclude that
\[ \beta^{n+1} \leq (1 + \tau_n)\beta^n \leq \cdots \leq \beta^0 \prod_{n_0=0}^{\infty} (1 + \tau_n) = T_\tau beta^0 =: \bar{\beta}, \]

implying that \( \{\beta^n\} \) is also uniformly bounded from above. Hence, the whole algorithm is well-defined.

We have the following lemma for the sequence generated by the algorithm.

**Lemma 3.1.** The sequence \( \{x^n\} \) generated by Algorithm 3.1 is bounded. Moreover, the sequence \( \{D(x^n)\} \) is monotonically decreasing and \( \lim_{n \to \infty} \|x^{n+1} - x^n\| = 0. \)

**Proof.** The boundedness of \( \{x^n\} \) follows trivially from the fact that \( \{x^n\} \subset B_R. \) We now pay our attention to the other two assertions.

Recall that \( x^{n+1} \) given by (29) is the unique minimizer of \( \Phi_{\beta^{n+1}}(w, x^n) \) on \( B_R. \) Therefore,
\[ \Phi_{\beta^{n+1}}(x^{n+1}, x^n) \leq \Phi_{\beta^{n+1}}(x^n, x^n), \]
i.e.,
\[ \|F(x^{n+1}) - y\|^2 - \|F(x^{n+1}) - F(x^n)\|^2 + \frac{r}{\beta^{n+1}} \|x^{n+1} - x^n\|^2 \leq \|F(x^n) - y\|^2 - \|F(x^n) - F(x^n)\|^2 + \frac{r}{\beta^{n+1}} \|x^n - x^n\|^2. \]

Rearranging terms, we obtain
\[ \begin{align*}
D(x^{n+1}) &= \|F(x^{n+1}) - y\|^2 \\
&\leq \|F(x^n) - y\|^2 - \frac{r}{\beta^{n+1}} \|x^{n+1} - x^n\|^2 + \|F(x^{n+1}) - F(x^n)\|^2 \\
&\leq D(x^n) - \frac{r}{\beta^{n+1}} \|x^{n+1} - x^n\|^2 + \frac{(1 - \delta)r}{\beta^{n+1}} \|x^{n+1} - x^n\|^2 \\
&= D(x^n) - \frac{\delta r}{\beta^{n+1}} \|x^{n+1} - x^n\|^2 \\
&\leq D(x^n) - \frac{\delta r}{\beta} \|x^{n+1} - x^n\|^2,
\end{align*} \]

where the second inequality follows from (30) and the last one from \( \beta^{n+1} \leq \bar{\beta}. \) Thus, \( \{D(x^n)\} \) is monotonically decreasing.

Rearranging terms yields
\[ \frac{\delta}{\beta} \|x^{n+1} - x^n\|^2 \leq D(x^n) - D(x^{n+1}). \]
Summing both sides over all $n$ further yields
\[ \sum_{n=0}^{\infty} \frac{R}{\beta} \|x^{n+1} - x^n\|^2 \leq \mathcal{D}(x^0) < \infty, \]
indicating that
\[ \lim_{n \to \infty} \|x^{n+1} - x^n\| = 0. \]
The proof is complete. \qed

3.2. Weak convergence. In this subsection, we prove the weak convergence of the generated sequence. Since $x^n \in B_R$, it has at least one weak accumulation point.

**Lemma 3.2.** If $x^*$ is a weak accumulation point of \{x^n\}, then it fulfills the necessary condition for a minimizer of $\mathcal{D}(x)$ on $B_R$, i.e.,
\[ (F'(x^*)^*(y - F(x^*)), w - x^*) \leq 0, \text{ for all } w \in B_R. \]

**Proof.** Recall that
\[ x^{n+1} = P_R \left( x^n + \frac{\beta_{n+1}}{r} F'(x^{n+1})^*(y - F(x^n)) \right), \]
or equivalently
\[ \langle x^n + \frac{\beta_{n+1}}{r} F'(x^{n+1})^*(y - F(x^n)) - x^{n+1}, w - x^{n+1} \rangle \leq 0 \text{ for all } w \in B_R. \]
Since $\|x^n - x^{n+1}\| \to 0$ and $\{\beta_n\} \subseteq [\underline{\beta}, \overline{\beta}] \subset (0, \infty)$, it follows from the above inequality that
\[ \limsup_{n \to \infty} \langle F'(x^{n+1})^*(y - F(x^n)), w - x^{n+1} \rangle \leq 0 \text{ for all } w \in B_R. \]
The result $\|x^n - x^{n+1}\| \to 0$ and the continuity of $F'$ imply that
\[ |\langle (F'(x^n)^* - F'(x^{n+1})^*)(y - F(x^n)), w - x^{n+1} \rangle| \leq \|F'(x^n) - F'(x^{n+1})\| \|y - F(x^n)\| \|w - x^{n+1}\| \to 0, \text{ as } n \to \infty, \]
and
\[ |\langle F'(x^n)^*(y - F(x^n)), x^n - x^{n+1} \rangle| \leq \|F'(x^n)\| \|y - F(x^n)\| \|x^n - x^{n+1}\| \to 0, \text{ as } n \to \infty. \]
Combining (34)-(36), we have
\[ \limsup_{n \to \infty} \langle F'(x^n)^*(y - F(x^n)), w - x^n \rangle \leq 0 \text{ for all } w \in B_R. \]
Since $x^*$ is a weak accumulation point of \{x^n\}, there is a subsequence, e.g., \{x^{n_j}\} of \{x^n\} converging weakly to $x^*$. Hence,
\[ F'(x)x^{n_j} \rightharpoonup F'(x)x^*. \]
Specially, (37) holds for this subsequence, i.e.,
\[ \limsup_{j \to \infty} \langle F'(x^{n_j})^*(y - F(x^{n_j})), w - x^{n_j} \rangle \leq 0 \text{ for all } w \in B_R. \]
The rest proof is the same as that of Proposition 9 in [21] and we omit it here. \qed
3.3. Strong convergence. The strong convergence can be proved in the same way as that of [21]; see Section 3.3 of [21].

**Theorem 3.3.** With the same assumptions as in Lemma 3.2 and assumptions (9) and (10) on the nonlinear operator \( F \), there exists a subsequence \( \{x^{n_j}\} \) such that it converges in norm toward the weak accumulation point \( x^* \), i.e.

\[
\lim_{n_j \to \infty} \|x^{n_j} - x^*\| = 0.
\]

**Proof.** Denote by \( \{x^n\} \) the subsequence that was introduced in the proof of Lemma 3.2 and let \( \{\beta^n\} \) be a subsequence of step size. Define now \( u_l = x^{n_l} - x^* \), \( v_l = x^{n_l+1} - x^* \) and \( \beta_l := \beta^{n_l+1} \). From the conclusion of Lemma 3.1, we have that \( \lim_{l \to \infty} \|u_l - v_l\| = 0 \). On the other hand,

\[
u_l - v_l = u_l + x^* - P_R(u_l + x^* + \beta_l F'(v_l + x^*)(y - F(u_l + x^*))) = u_l + P_R(x^* + \beta_l F'(x^*)(y - F(x^*))) - P_R(u_l + x^* + \beta_l F'(v_l + x^*)(y - F(u_l + x^*))) = u_l + P_R(x^* + \beta_l F'(x^*)(y - F(x^*))) - P_R(x^* + \beta_l F'(v_l + x^*)(y - F(u_l + x^*)) + u_l) + P_R(x^* + \beta_l F'(x^*)(y - F(x^*))) + u_l - P_R(x^* + \beta_l F'(x^*)(y - F(x^*))) + u_l
\]

where we have applied Lemma 3.2 and Lemma 2.2. By the assumptions on \( F \) and since the \( \beta_l \) are uniformly bounded, we obtain

\[
\|P_R(x^* + \beta_l F'(x^*)(y - F(x^*))) + u_l\| \leq \|\beta_l F'(x^*)(y - F(u_l + x^*) + u_l)\|
\]

\[
\leq \|\beta_l F'(x^*)(y - F(x^*))\| \sup_{x \in B_R} \|F'(x)\| \|F(u_l + x^*) - F(x^*)\|
\]

\[
\to 0,
\]

(38)

where the last expression is due to (8). Similar, from (8)-(10) we have

\[
\|P_R(x^* + \beta_l F'(x^*)(y - F(u_l + x^*)) + u_l) - P_R(x^* + \beta_l F'(v_l + x^*)(y - F(u_l + x^*)) + u_l)\|
\]

\[
\leq \|\beta_l F'(x^*)(y - F(v_l + x^*)) + u_l\| \|F'(x^*) - F'(v_l + x^*)\| \|y - F(x^*)\| + L\|u_l\| \|F(x^*) - F(u_l + x^*)\|
\]

\[
\leq \beta \|F'(x^*) - F'(v_l + x^*)\| \|y - F(x^*)\| + L\|u_l\| \|F(x^*) - F(u_l + x^*)\|
\]

\[
\to 0.
\]

Consequently, combining \( \|u_l - v_l\| \to 0 \) and the two last statements, we observe that

\[
\lim_{l \to \infty} \|P_R(x^* + \beta_l F'(x^*)(y - F(x^*)) + u_l) - P_R(x^* + \beta_l F'(x^*)(y - F(x^*))) - u_l\| = 0.
\]

Since the sequence \( \beta_l \) is bounded, there must be at least one accumulation point, which we denote by \( \beta^\infty \). We choose a subsequence \( \{\beta_j\} \in N \), such that

\[
\lim_{j \to \infty} \beta_j = \beta^\infty,
\]
and
\[
\lim_{j \to \infty} \| P_R(x^* + \beta_j F'(x^*)(y - F(x^*)) + u_{ij}) - P_R(x^* + \beta_j F'(x^*)(y - F(x^*))) - u_{ij} \| = 0.
\]
Denote \( h^* = x^* + \beta^\infty F'(x^*)(y - F(x^*)) \), so we have
\[
\lim_{j \to \infty} \| P_R(h^* + u_{ij}) - P_R(h^*) - u_{ij} \| = 0.
\]
Noticing that for every \( h^* + u_{ij} \), there exists \( \mu_j \geq 0 \), such that
\[
P_R(h^* + u_{ij}) = S_{\mu_j} (h^* + u_{ij}) = h^* + u_{ij} - \min \{ \mu_j, \| h^* + u_{ij} \| \} \frac{h^* + u_{ij}}{\| h^* + u_{ij} \|}.
\]
Let
\[
\sigma(l_j) = 1 - \frac{\min \{ \mu_j, \| h^* + u_{ij} \| \}}{\| h^* + u_{ij} \|},
\]
so \( P_R(h^* + u_{ij}) = \sigma(l_j) h^* + \sigma(l_j) u_{ij} \). By the same definition, there exists \( \mu \geq 0 \), such that \( P_R(h^*) = \mu h^* \). Without loss of generality, we can assume \( \| h^* \| > \rho \), so \( \mu < 1 \).
Let \( \| h^* \|^{1-\mu} \geq \epsilon > 0 \). Because \( \| h^* \| \) is bounded, there exists \( K > 0 \), such that \( \sum_{\lambda > K} |h^*| \lambda \leq \frac{\| h^* \|}{2} \), and due to the weak convergence of \( \{ u_{nj} \} \), there exists \( N_1 > 0 \), such that when \( n_j \geq N_1 \), \( \sum_{\lambda \leq K} |u_{nj}| \lambda \leq \epsilon \). For the same \( \epsilon \), from (39) there exists \( N_2 > 0 \), when \( n_j \geq N_2 \),
\[
\epsilon \geq \sum_{\lambda \leq K} |\sigma(n_j) h^* + \sigma(n_j) u_{nj} - \mu h^* - u_{nj}| \lambda
\]
\[
+ \sum_{\lambda > K} |\sigma(n_j) h^* + \sigma(n_j) u_{nj} - \mu h^* - u_{nj}| \lambda.
\]
So when \( n_j \geq \max(N_1, N_2) \), we have
\[
\epsilon \geq \sum_{\lambda \leq K} |\sigma(n_j) h^* + \sigma(n_j) u_{nj} - \mu h^* - u_{nj}| \lambda
\]
and
\[
\epsilon \geq \sum_{\lambda > K} |\sigma(n_j) h^* + \sigma(n_j) u_{nj} - \mu h^* - u_{nj}| \lambda.
\]
From (40), we have
\[
\epsilon \geq \sum_{\lambda \leq K} |\sigma(n_j) h^* + \sigma(n_j) u_{nj} - \mu h^* - u_{nj}| \lambda
\]
\[
\geq \sum_{\lambda \leq K} |(\sigma(n_j) - \mu) h^*| \lambda - \sum_{\lambda \leq K} |(\sigma(n_j) - 1) u_{nj}| \lambda.
\]
Because
\[
\sum_{\lambda \leq K} |(\sigma(n_j) - 1) u_{nj}| \lambda = |\sigma(n_j) - 1| \sum_{\lambda \leq K} |u_{nj}| \lambda \leq \epsilon,
\]
and
\[
\sum_{\lambda \leq K} |h^*| \lambda = \| h^* \| - \sum_{\lambda > K} |h^*| \lambda \geq \frac{\| h^* \|}{2},
\]
we deduce
\[
\frac{4}{\| h^* \|} \epsilon \geq |\sigma(n_j) - \mu|,
\]
and as a consequence \( \frac{1-\mu}{2} \leq 1 - |\sigma(n_j) - \frac{3(1-\mu)}{2}|. \)
we have

\[ f \quad \text{and then sampled in a non-ideal manner. In [21], the authors specified the model problem in [21]. This example comes from sensing setup in which a continuous-improvements over the latter one. The test problem here is a nonlinear sampling algorithm, and compare it with the method in [21] (i.e., Algorithm 1.1) to show its\]

4. **Numerical experiments.** In this section, we test the performance of the new algorithm, and compare it with the method in [21] (i.e., Algorithm 1.1) to show its improvements over the latter one. The test problem here is a nonlinear sampling problem in [21]. This example comes from sensing setup in which a continuous-time signal is mapped by a memoryless, invertible and nonlinear transformation, and then sampled in a non-ideal manner. In [21], the authors specified the model to reconstruct \( f^*(t) \) by the transformed function \( y = S \circ M \circ f^*(t) \):

\[
\min_{x} \|F(x) - y\|^2 \\
\text{s.t. } x \in B_R,
\]

where \( F(x) = S \circ M \circ A^*(x), A^* : x \rightarrow \sum_{m \in Z_1} x_m a_m(t), M : f \rightarrow \sqrt{f^2 + \varepsilon^2}, S : f \rightarrow \{ (f, s_n(t)) \}_{n \in Z_2}, a_m(t) = \frac{1}{\sqrt{T_m}} \text{sinc}(\frac{\pi}{T_m}(t - mT_m)), s_n(t) = \frac{1}{\sqrt{T_n}} \text{sinc}(\frac{\pi}{T_n}(t - nT_n)), \langle a(t), b(t) \rangle = \int_R a(t)b(t)dt. \)

In this example, we try to reconstruct \( f^*(t) = a_{-2}(t) - 0.5a_{2.5}(t) \). We set \( Z_1 = \{0, 1, 2, \ldots, m - 1\} \) and \( Z_2 = \{0, 1, 2, \ldots, n - 1\} \). The larger \( m \) and \( n \) mean the more precise description of \( f^* \) and \( y \) respectively, so we choose different \( m, n \) to show the efficiency of our algorithm in different cases.

We also reconstruct the inaccurate \( f^* \), which is obtained by measurements that contain noise. We set the noisy data \( f^*(t) = a_5(t) - 0.5a_1(t) + e \), where \( e = \text{rand(size}(f^*(t))) \).

We set the stopping rule as \( \|e(x^k, \beta)\|^2 \leq 10^{-5} \). The parameters of the algorithms are set as: \( D_0 = 0.4, D_n = 0.2, T_n = 0.1, T_2 = 0.1, \tau_n = 0.9, \delta = 0.2, \mu = 0.5, \rho = 0.5, x_0 = \text{rand}(m, 1), x_0(1) = 1, x_0(5) = -1, \beta_0 = 1 \).

The radius \( R \) plays a key role in the model, as the regularization parameter \( \alpha \) in (2). However, both \( R \) and \( \alpha \) are quite difficult to choose in applications. We hence first settle down it before we compare the algorithms. For this end, we set \( m, n = 10 \) and choose different values of \( R: R = 1.0, 1.3, 1.5, 1.6, 1.7, 1.8, \) and 2.0.
The running time \((\text{time})\) of the algorithm and the number of iterations \((\text{iter.})\) are listed in Table 1. From Table 1 we can observe that for \(R\) in the interval \([1.3, 1.8]\), the results are better and we set \(R = 1.6\) in the following test. To justify this, we also give the corresponding results when \(R = 1.0, 1.5, 1.8, 2.0\) in Figure 1.

### Table 1. Numerical results with different values of \(R\)

| \(m,n=10\) | Algorithm 3.1 | Algorithm 1.1 |
|------------|---------------|---------------|
| \(R\)     | time(s)       | iter.         |
| 1.0        | 30.87         | 15            |
| 1.3        | 47.09         | 14            |
| 1.5        | 28.29         | 26            |
| 1.6        | 24.84         | 19            |
| 1.7        | 59.49         | 54            |
| 1.8        | 56.64         | 32            |
| 2.0        | 91.79         | 69            |

Case 1. The obtained data is accurate. First, we show the case in which the data \(y\) is accurate. In Table 2, we report the running time and the number of iterations for \(n\) and \(m\) varying from 5 to 160. From this table we can see that for all cases, to achieve the same accuracy, our algorithm needs less number of iterations and less CPU time than Algorithm 1.1.

We now illustrate and compare the two algorithms in Figure 2 by plotting \(\beta_k\) and the residual of the problem at each iteration in Algorithm 3.1 and Algorithm 1.1.

The first figures of all subfigures in Figure 2 show that \(\beta_k\) in our algorithm is almost larger than \(\beta_k\) in Algorithm 1.1. Just as we mentioned before, \(\beta_k\) could be
Figure 2. The value of $\beta^k$ and the residual of the problem at each iteration in Algorithm 3.1 and Algorithm 1.1, in case of (a) $m, n = 5$; (b) $m, n = 10$; (c) $m, n = 20$; (d) $m, n = 40$; (e) $m, n = 80$; (f) $m, n = 160$. 
Table 2. Numerical results of for the accurate case

| (m,n)  | Algorithm 3.1 | Algorithm 1.1 |
|--------|--------------|--------------|
|        | time(s) | iter. | time(s) | iter. |
| (5,5)  | 13.80 | 17     | 40.74  | 57    |
| (10,10)| 26.82 | 16     | 91.76  | 64    |
| (20,20)| 81.61 | 32     | 134.30 | 58    |
| (40,40)| 108.04| 14     | 480.42 | 69    |
| (80,80)| 355.52| 32     | 882.87 | 67    |
| (160,160)| 843.62| 36     | 2326.32| 79    |

considered as the stepsize in every iteration, and the larger $\beta^k$ means the larger probability of the high speed to the convergent point. In fact, our algorithm is faster than Algorithm 1.1, which shows our algorithm is better. The second figures of all subfigures in Figure 2 show the residual of the problem at each iteration in Algorithm 3.1 and Algorithm 1.1. From Figure 2 and Table 2, it is easy to see that our algorithm has the same result as Algorithm 1.1, and just like our theoretical analysis, the behavior of our algorithm is better than another one, which only needs less than half iterations of Algorithm 1.1, and needs less CPU time to achieve the same accuracy.

In our experiment, we use the iterative method to get $x^{n+1}$, the computing of it in every iteration is expensive. We present the inner iteration numbers in every iteration in Figure 3. From that figure, we could see that the average inner iteration number of our algorithm is not too difference from Algorithm 1.1. But the outer iteration number of our algorithm is smaller, this is another point explains that our algorithm is faster.

In Figure 4, we will show the signal which is reconstructed by Algorithm 3.1 and Algorithm 1.1 in comparison with the signal we obtained. We find that the signal reconstructed by our algorithm is the similar to the signal reconstructed by Algorithm 1.1.

One major relaxation of of Algorithm 3.1 is the use of a new step size rule that can provide higher possibility to use a larger step size, which can reduce the number of iterations of the algorithm. However, the larger of the initial guess of the step size, the slower the convergence of the ‘inner’ fixed point iteration. Nevertheless, since the inner fixed point iteration of Algorithm 3.1 is also in a self-adaptive manner, the inner number of iterations of Algorithm 3.1 is usually less than the method with constant step size too, as shown in Figure 3.

Case2. The obtained data is inaccurate. Now we will illustrate the case the observed data $y$ contains noise. Figure 5 shows the value of $\beta^k$ and the residual of the problem at each iteration in Algorithm 3.1 and Algorithm 1.1, when the data which we need to be reconstructed contains noise. Figure 5 also shows that $\beta^k$ of our algorithm is almost larger than $\beta^k$ in Algorithm 1.1, which means our algorithm is faster than Algorithm 1.1.

From Table 3 and Figure 5, it is easy to see that our algorithm is still faster than Algorithm 1.1, when we have the same stopping rule in both algorithms.

We also present the inner iteration number of the computing of $x^{n+1}$ in every iteration in Figure 6. From this figure, we could see that the average inner iteration number is not too different from Algorithm 1.1. Since the outer iteration number of our algorithm is smaller, that is another point explains that our algorithm is faster.
Figure 3. The number of inner iteration in Algorithm 3.1 and Algorithm 1.1, in case of (a) \(m, n = 5\); (b) \(m, n = 10\); (c) \(m, n = 20\); (d) \(m, n = 40\); (e) \(m, n = 80\); (f) \(m, n = 160\); without noise.

Figure 4. The signal which is reconstructed by Algorithm 3.1 and Algorithm 1.1, in case of (a) \(m, n = 5\); (b) \(m, n = 10\); (c) \(m, n = 20\); (d) \(m, n = 40\); (e) \(m, n = 80\); (f) \(m, n = 160\).
Figure 5. The value of $\beta^k$ and the residual of the problem at each iteration in Algorithm 3.1 and Algorithm 1.1, in case of (a) $m, n = 5$; (b) $m, n = 10$; (c) $m, n = 20$; (d) $m, n = 40$; (e) $m, n = 80$; (f) $m, n = 160$. 

Inverse Problems and Imaging Volume 10, No. 3 (2016), 689–709
Table 3. Numerical results for data with noise

| (m,n)   | Algorithm 3.1 time(s) | Algorithm 3.1 iter. | Algorithm 1.1 time(s) | Algorithm 1.1 iter. |
|---------|-----------------------|---------------------|-----------------------|---------------------|
| (5,5)   | 11.43                 | 13                  | 61.06                 | 98                  |
| (10,10) | 22.73                 | 16                  | 177.95                | 168                 |
| (20,20) | 189.26                | 13                  | 828.33                | 384                 |
| (40,40) | 384.78                | 18                  | 1426.47               | 368                 |
| (80,80) | 917.77                | 19                  | 3472.48               | 401                 |
| (160,160)| 2790.11               | 22                  | 12276.44              | 712                 |

Figure 6. The number of inner iteration in Algorithm 3.1 and Algorithm 1.1, in case of (a) m, n = 5; (b) m, n = 10; (c) m, n = 20; (d) m, n = 40; (e) m, n = 80; (f) m, n = 160; with noise.

In Figure 7, we will show the signal which is reconstructed by Algorithm 3.1 and Algorithm 1.1 in comparison with the signal we observed with noise and without noise. We find that the signal reconstructed by our algorithm is better than the signal reconstructed by Algorithm 1.1.

5. Conclusions. In this paper, we proposed an efficient projection method for solving the nonlinear inverse problems with sparsity constraint. Under the condition that the operator $F$ be continuously Fréchet differentiable, a weaker condition than that in Teschke and Borries [21] which required $F$ to be twice Fréchet differentiable, we managed to prove the convergence of the algorithm. Such a weaker requirement enables our algorithm to be applied in wider fields. Moreover, the factor in our application criterion of $\beta^n$ is $1 - \delta$, while it is $1/2$ in [21]. This slight change makes it possible to accept a larger $\beta^n$, which may effect the algorithm’s efficiency. From
Figure 7. The signal which is reconstructed by Algorithm 3.1 and Algorithm 1.1, in case of (a) \( m, n = 5 \); (b) \( m, n = 10 \); (c) \( m, n = 20 \); (d) \( m, n = 40 \); (e) \( m, n = 80 \); (f) \( m, n = 160 \).

the numerical point of view, we select the initial guess using the information from the last iteration and more importantly, we adopt a ‘self-adaptive’ procedure in selecting the next trial of \( \beta^n \), i.e., Step S3 in the algorithm. In the numerical experiments, we tested the algorithm, not only for the exact observed data, but also for the data with noise. Our results show that both our algorithm and that of Teschke and Borries [21] are quite robust to the noise.

Acknowledgments. The research is supported by a project funded by PAPD of Jiangsu Higher Education Institutions and the NSFC grants 11371197, 11431002, 11271196.

REFERENCES

[1] J. F. Bonnans and A. Shapiro, Perturbation Analysis of Optimization Problems, Springer-Verlag, New York, 2000.
[2] K. Bredies, D. A. Lorenz and P. Maass, A generalized conditional gradient method and its connection to an iterative shrinkage method, Comput. Optim. Appl., 42 (2009), 173–193.
[3] X. J. Cai, G. Y. Gu and B. S. He, A proximal point algorithm revisit on the alternating direction method of multipliers, Sci. China Math., 56 (2013), 2179–2186.
[4] I. Daubechies, M. Defrise and C. DeMol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Commun. Pure Appl. Math., 57 (2004), 1413–1457.
[5] I. Daubechies, G. Teschke and L. Vese, Iteratively solving linear inverse problems with sparsity constraints, Inverse Problems Imag., 1 (2007), 29–46.
[6] I. Daubechies, G. Teschke and L. Vese, Iteratively solving linear inverse problems with general convex constraints, Inverse Problems Imag., 1 (2007), 29–46.
[7] I. Daubechies, G. Teschke and L. Vese, On some iterative concepts for image restoration, Adv. Imag. Electron Phys., 150 (2008), 1–51.
[8] Y. C. Eldar, T. G. Dvorkind and E. Matusiak, Nonlinear and non-ideal sampling: Theory and methods, IEEE Trans. Signal Process., 56 (2008), 5874–5890.
[9] H. W. Engl, M. Hanke and A. Neubauer, Regularization of Inverse Problems, Springer, Netherlands, 1996.
[10] F. Facchinei and J. S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Volumes I and II, Springer-Verlag, Berlin, 2003.

[11] M. Fornasier, Domain decomposition methods for linear inverse problems with sparsity constraints, *Inverse Problems*, 23 (2007), 2505–2526.

[12] 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), 586–597.

[13] M. Fornasier and H. Rauhut, Recovery algorithms for vector valued data with joint sparsity constraint, *SIAM J. Numer. Anal.*, 46 (2008), 577–613.

[14] Z. L. Ge, G. Qian and D. R. Han, Global convergence of an inexact operator splitting method for monotone variational inequalities, *J. Ind. Man. Optim.*, 7 (2011), 1013–1026.

[15] B. S. He, H. Yang, Q. Meng and D. R. Han, Modified Goldstein-Levitin-Polyak projection method for asymmetric strong monotone variational inequality, *J. Optim. Theory Appl.*, 112 (2002), 129–143.

[16] T. Hein and K. S. Kazimierski, Accelerated Landweber iteration in Banach spaces, *Inverse Problems*, 26 (2010), 055002 (17pp).

[17] B. Kaltenbach, F. Schöpfer and T. Schuster, Iterative methods for nonlinear ill-posed problems in Banach spaces: convergence and applications to parameter identification problems, *Inverse Problems*, 25 (2009), 065003 (19pp).

[18] R. Ramlau and G. Teschke, Tikhonov replacement functionals for iteratively solving nonlinear operator equations, *Inverse Problems*, 21 (2005), 1571–1592.

[19] R. Ramlau and G. Teschke, A Tikhonov-based projection iteration for nonlinear Ill-posed problems with sparsity constraints, *Numer. Math.*, 104 (2006), 177–203.

[20] G. Teschke, Multi-frame representations in linear inverse problems with mixed multi-constraints, *Appl. Comput. Harmon. Anal.*, 22 (2007), 43–60.

[21] G. Teschke and C. Borries, Accelerated projected steepest descent method for nonlinear inverse problems with sparsity constraints, *Inverse Problems*, 26 (2010), 025007 (23pp).

[22] E. van den Berg and M. P. Friedlander, Probing the Pareto frontier for basic pursuit solutions, *SIAM J. Sci. Comput.*, 31 (2008), 890–912.

[23] E. Zeidler, *Nonlinear Functional Analysis and Its Applications III*, Springer, New York, 1985.

[24] E. Zeidler, *Nonlinear Functional Analysis and Its Applications I*, Springer, New York, 1986.

[25] W. X. Zhang, D. R. Han and Z. B. Li, A self-adaptive projection method for solving the multiple-sets split feasibility problem, *Inverse Problems*, 25 (2009), 115001 (16pp).

[26] T. Zhu and Z. Yu, A simple proof for some important properties of the projection mapping, *Mathematical Inequalities and Applications*, 7 (2004), 453–456.

Received July 2015; revised March 2016.

*E-mail address*: handeren@njnu.edu.cn

*E-mail address*: jiazehui90@126.com

*E-mail address*: yzsong@njnu.edu.cn

*E-mail address*: wangzhiwei@ntu.edu.sg