

A REGULARIZED SEMI-SMOOTH NEWTON METHOD WITH PROJECTION STEPS FOR COMPOSITE CONVEX PROGRAMS*

XIANTAO XIAO[†], YONGFENG LI[‡], ZAIWEN WEN[§], AND LIWEI ZHANG[¶]

Abstract. The goal of this paper is to study approaches to bridge the gap between first-order and second-order type methods for composite convex programs. Our key observations are: i) Many well-known operator splitting methods, such as forward-backward splitting (FBS) and Douglas-Rachford splitting (DRS), actually define a fixed-point mapping; ii) The optimal solutions of the composite convex program and the solutions of a system of nonlinear equations derived from the fixed-point mapping are equivalent. **Solving this kind of system of nonlinear equations enables us to develop second-order type methods.** Although these nonlinear equations may be non-differentiable, they are often semi-smooth and their generalized Jacobian matrix is positive semidefinite due to monotonicity. By combining with a regularization approach and a known hyperplane projection technique, we propose an adaptive semi-smooth Newton method and establish its convergence to global optimality. Preliminary numerical results on ℓ_1 -minimization problems demonstrate that our second-order type algorithms are able to achieve superlinear or quadratic convergence.

Key words. composite convex programs, operator splitting methods, proximal mapping, semi-smoothness, Newton method

AMS subject classifications. 90C30, 65K05

1. Introduction. This paper aims to solve a composite convex optimization problem in the form

$$(1.1) \quad \min_{x \in \mathbb{R}^n} f(x) + h(x),$$

where f and h are extended real-valued convex functions. Problem (1.1) arises from a wide variety of applications, such as signal recovery, image processing, machine learning, data analysis, and etc. For example, it becomes the sparse optimization problem when f or h equals to the ℓ_1 -norm, which attracts a significant interest in signal or image processing in recent years. If f is a loss function associated with linear predictors and h is a regularization function, problem (1.1) is often referred as the regularized risk minimization problem in machine learning and statistics. When f or h is an indicator function onto a convex set, problem (1.1) represents a general convex constrained optimization problem.

Recently, a series of first-order methods, including the forward-backward splitting (FBS) (also known as proximal gradient) methods, Nesterov's accelerated methods, the alternative direction methods of multipliers (ADMM), the Douglas-Rachford splitting (DRS) and Peaceman-Rachford splitting (PRS) methods, have been extensively studied and widely used for solving a subset of problem (1.1). The readers are referred to, for example, [3, 6] and references therein, for a review on some of these first-order methods. One main feature of these methods is that they first exploit the underlying problem structures, then construct subproblems that can be solved relatively efficiently. These algorithms are rather simple yet powerful

*The first version was titled "Semi-Smooth Second-order Type Methods for Composite Convex Programs" (<http://arxiv.org/abs/1603.07870>). This version was submitted to "SIAM Journal on Optimization" on July 28, 2016.

[†]School of Mathematical Sciences, Dalian University of Technology, Dalian, CHINA (xtxiao@dlut.edu.cn). Research supported by the Fundamental Research Funds for the Central Universities under the grant DUT16LK30.

[‡]School of Mathematical Sciences, Peking University, Beijing, CHINA (1200010733@pku.edu.cn).

[§]Beijing International Center for Mathematical Research, Peking University, Beijing, CHINA (wenzw@pku.edu.cn). Research supported in part by NSFC grant 11322109 and by the National Basic Research Project under the grant 2015CB856002.

[¶]School of Mathematical Sciences, Dalian University of Technology, Dalian, CHINA (lwzhang@dlut.edu.cn). Research partially supported by NSFC grants 11571059 and 91330206.

since they are easy to be implemented in many interested applications and they often converge fast to a solution with moderate accuracy. However, a notorious drawback is that they may suffer from a slow tail convergence and hence a significantly large number of iterations is needed in order to achieve a high accuracy.

A few Newton-type methods for some special instances of problem (1.1) have been investigated to alleviate the inherent weakness of the first-order type methods. Most existing Newton-type methods for problem (1.1) with a differentiable function f and a simple function h whose proximal mapping can be cheaply evaluated are based on the FBS method to some extent. The proximal Newton method [19, 28] can be interpreted as a generalization of the proximal gradient method. It updates in each iteration by a composition of the proximal mapping with a Newton or quasi-Newton step. The semi-smooth Newton methods proposed in [16, 24, 4] solve the nonsmooth formulation of the optimality conditions corresponding to the FBS method. In [42], the augmented Lagrangian method is applied to solve the dual formulation of general linear semidefinite programming problems, where each augmented Lagrangian function is minimized by using the semi-smooth Newton-CG method. Similarly, a proximal point algorithm is developed to solve the dual problems of a class of matrix spectral norm approximation in [5], where the subproblems are again handled by the semi-smooth Newton-CG method.

In this paper, we study a few second-order type methods for problem (1.1) in a general setting even if f is nonsmooth and h is an indicator function. Our key observations are that many first-order methods, such as the FBS and DRS methods, can be written as fixed-point iterations and the optimal solutions of (1.1) are also the solutions of a system of nonlinear equations defined by the corresponding fixed-point mapping. Consequently, the concept is to develop second-order type algorithms based on solving the system of nonlinear equations. Although these nonlinear equations are often non-differentiable, they are monotone and can be semi-smooth due to the properties of the proximal mappings. We first propose a regularized semi-smooth Newton method to solve the system of nonlinear equations. The regularization term is important since the generalized Jacobian matrix corresponding to monotone equations may only be positive semidefinite. In particular, the regularization parameter is updated by a self-adaptive strategy similar to the trust region algorithms. By combining with the semi-smooth Newton step and a hyperplane projection technique, we show that the method converges globally to an optimal solution of problem (1.1). The hyperplane projection step is in fact indispensable for the convergence to global optimality and it is inspired by several iterative methods for solving monotone nonlinear equations [34, 45]. Different from the approaches in the literature, the hyperplane projection step is only executed when the residual of the semi-smooth Newton step is not reduced sufficiently. When certain conditions are satisfied, we prove that the semi-smooth Newton steps are always performed close to the optimal solutions. Consequently, fast local convergence rate is established. For some cases, the computational cost can be further reduced if the Jacobian matrix is approximated by the limited memory BFGS (L-BFGS) method.

Our main contribution is the study of some relationships between the first-order and second-order type methods. Our semi-smooth Newton methods are able to solve the general convex composite problem (1.1) as long as a fixed-point mapping is well defined. In particular, our methods are applicable to constrained convex programs, such as constrained ℓ_1 -minimization problem. In contrast, the Newton-type methods in [19, 28, 16, 24, 4] are designed for unconstrained problems. Unlike the methods in [42, 5] applying the semi-Newton method to a sequence of subproblems, our target is a single system of nonlinear equations. Although solving the Newton system is a major challenge, the computational cost usually can be controlled reasonably well when certain structures can be utilized. Our preliminary

numerical results show that our proposed methods are able to reach superlinear or quadratic convergence rates on typical ℓ_1 -minimization problems.

The rest of this paper is organized as follows. In section 2, we review a few popular operator splitting methods, derive their equivalent fixed-point iterations and state their convergence properties. We propose a semi-smooth Newton method and establish its convergence results in section 3. Numerical results on a number of applications are presented in section 4. Finally, we conclude this paper in section 5.

1.1. Notations. Let I be the identity operator or identity matrix of suitable size. Given a convex function $f : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ and a scalar $t > 0$, the *proximal mapping* of f is defined by

$$(1.2) \quad \mathbf{prox}_{tf}(x) := \arg \min_{u \in \mathbb{R}^n} f(u) + \frac{1}{2t} \|u - x\|_2^2.$$

If $f(x) = 1_\Omega(x)$ is the indicator function of a nonempty closed convex set $\Omega \subset \mathbb{R}^n$, then the proximal mapping \mathbf{prox}_{tf} reduces to the *metric projection* defined by

$$(1.3) \quad \mathcal{P}_\Omega(x) := \arg \min_{u \in \Omega} \frac{1}{2} \|u - x\|_2^2.$$

The Fenchel conjugate function f^* of f is

$$(1.4) \quad f^*(y) := \sup_{x \in \mathbb{R}^n} \{x^T y - f(x)\}.$$

A function f is said to be *closed* if its epigraph is closed, or equivalently f is lower semicontinuous. A mapping $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is said to be *monotone*, if

$$\langle x - y, F(x) - F(y) \rangle \geq 0, \quad \text{for all } x, y \in \mathbb{R}^n.$$

2. Operator splitting and fixed-point algorithms. This section reviews some operator splitting algorithms for problem (1.1), including FBS, DRS, and ADMM. These algorithms are well studied in the literature, see [12, 2, 6, 7] for example. Most of the operator splitting algorithms can also be interpreted as fixed-point algorithms derived from certain optimality conditions.

2.1. FBS. In problem (1.1), let h be a continuously differentiable function. The FBS algorithm is the iteration

$$(2.1) \quad x^{k+1} = \mathbf{prox}_{tf}(x^k - t\nabla h(x^k)), \quad k = 0, 1, \dots,$$

where $t > 0$ is the step size. When $f(x) = 1_C(x)$ is the indicator function of a closed convex set C , FBS reduces to the *projected gradient method* for solving the constrained program

$$\min_{x \in \mathbb{R}^n} h(x) \quad \text{subject to} \quad x \in C.$$

Define the following operator

$$(2.2) \quad T_{\text{FBS}} := \mathbf{prox}_{tf} \circ (I - t\nabla h).$$

Then FBS can be viewed as a fixed-point iteration

$$(2.3) \quad x^{k+1} = T_{\text{FBS}}(x^k).$$

2.2. DRS. The DRS algorithm solves (1.1) by the following update:

$$(2.4) \quad x^{k+1} = \mathbf{prox}_{th}(z^k),$$

$$(2.5) \quad y^{k+1} = \mathbf{prox}_{tf}(2x^{k+1} - z^k),$$

$$(2.6) \quad z^{k+1} = z^k + y^{k+1} - x^{k+1}.$$

The algorithm is traced back to [8, 21, 10] to solve partial differential equations (PDEs). The fixed-point iteration characterization of DRS is in the form of

$$(2.7) \quad z^{k+1} = T_{\text{DRS}}(z^k),$$

where

$$(2.8) \quad T_{\text{DRS}} := I + \mathbf{prox}_{tf} \circ (2\mathbf{prox}_{th} - I) - \mathbf{prox}_{th}.$$

2.3. Dual operator splitting and ADMM. Consider a linear constrained program:

$$(2.9) \quad \begin{array}{ll} \min_{x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{R}^{n_2}} & f_1(x_1) + f_2(x_2) \\ \text{subject to} & A_1 x_1 + A_2 x_2 = b, \end{array}$$

where $A_1 \in \mathbb{R}^{m \times n_1}$ and $A_2 \in \mathbb{R}^{m \times n_2}$. The dual problem of (2.9) is given by

$$(2.10) \quad \min_{w \in \mathbb{R}^m} d_1(w) + d_2(w),$$

where

$$d_1(w) := f_1^*(A_1^T w), \quad d_2(w) := f_2^*(A_2^T w) - b^T w.$$

Assume that f_1 is closed and strongly convex (which implies that ∇d_1 is Lipschitz [31, Proposition 12.60]) and f_2 is convex. The FBS iteration for the dual problem (2.10) can be expressed in terms of the variables in the original problem under the name alternating minimization algorithm, which is also equivalent to the fixed-point iteration

$$w^{k+1} = T_{\text{FBS}}(w^k).$$

Assume that f_1 and f_2 are convex. It is widely known that the DRS iteration for dual problem (2.10) is the ADMM [15, 14]. It is regarded as a variant of augmented Lagrangian method and has attracted much attention in numerous fields. A recent survey paper [3] describes the applications of the ADMM to statistics and machine learning. The ADMM is equivalent to the following fixed-point iteration

$$z^{k+1} = T_{\text{DRS}}(z^k),$$

where T_{DRS} is the DRS fixed-point mapping for problem (2.10).

2.4. Convergence of the fixed-point algorithms. We summarize the relationship between the aforementioned fixed-points and the optimal solution of problem (1.1), and review the existing convergence results on the fixed-point algorithms.

The following lemma is straightforward, and its proof is omitted.

LEMMA 2.1. *Let the fixed-point mappings T_{FBS} and T_{DRS} be defined in (2.2) and (2.8), respectively.*

- (i) Suppose that f is closed, proper and convex, and h is convex and continuously differentiable. A fixed-point of T_{FBS} is equivalent to an optimal solution to problem (1.1).
- (ii) Suppose that f and h are both closed, proper and convex. Let z^* be a fixed-point of T_{DRS} , then $\text{prox}_{th}(z^*)$ is an optimal solution to problem (1.1).

Error bound condition is a useful property for establishing the linear convergence of a class of first-order methods including the FBS method and ADMM, see [11, 22, 36, 18] and the references therein. Let X^* be the optimal solution set of problem (1.1) and $F(x) \in \mathbb{R}^n$ be a residual function satisfying $F(x) = 0$ if and only if $x \in X^*$. The definition of error bound condition is given as follows.

DEFINITION 2.2. *The error bound condition holds for some test set T and some residual function $F(x)$ if there exists a constant $\kappa > 0$ such that*

$$(2.11) \quad \text{dist}(x, X^*) \leq \kappa \|F(x)\|_2 \quad \text{for all } x \in T.$$

In particular, it is said that error bound condition with residual-based test set (EBR) holds if the test set in (2.11) is selected by $T := \{x \in \mathbb{R}^n | f(x) + h(x) \leq v, \|F(x)\|_2 \leq \varepsilon\}$ for some constant $\varepsilon \geq 0$ and any $v \geq v^* := \min_x f(x) + h(x)$.

Under the error bound condition, the fixed-point iteration of FBS is proved to converge linearly, see [9, Theorem 3.2] for example.

PROPOSITION 2.3 (Linear convergence of FBS). *Suppose that error bound condition (EBR) holds with parameter κ for residual function F_{FBS} . Let x^* be the limit point of the sequence $\{x^k\}$ generated by the fixed-point iteration $x^{k+1} = T_{FBS}(x^k)$ with $t \leq \beta^{-1}$ for some constant $\beta > 0$. Then there exists an index r such that for all $k \geq 1$,*

$$\|x^{r+k} - x^*\|_2^2 \leq \left(1 - \frac{1}{2\kappa\beta}\right)^k C \cdot (f(x^r) + h(x^r) - f(x^*) - h(x^*)),$$

where $C := \frac{2}{\beta(1 - \sqrt{1 - (2\beta\gamma)^{-1}})^2}$.

Finally, we mention that the sublinear convergence rate of some general fixed-point iterations has been well studied, see [7, Theorem 1].

3. Semi-smooth Newton method for nonlinear monotone equations. The purpose of this section is to design a Newton-type method for solving the system of nonlinear equations

$$(3.1) \quad F(z) = 0,$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is strongly semi-smooth and monotone. In particular, we are interested in $F(z) = z - T(z)$, where $T(z)$ is a fixed-point mapping corresponding to certain first-order type algorithms.

3.1. Semi-smoothness of proximal mapping. We now discuss the semi-smoothness of proximal mappings. This property often implies that the fixed-point mappings corresponding to operator splitting algorithms are semi-smooth or strongly semi-smooth.

Let $\mathcal{O} \subseteq \mathbb{R}^n$ be an open set and $F : \mathcal{O} \rightarrow \mathbb{R}^m$ be a locally Lipschitz continuous function. Rademacher's theorem says that F is almost everywhere differentiable. Let D_F be the set of differentiable points of F in \mathcal{O} . We next introduce the concepts of generalized differential.

DEFINITION 3.1. *Let $F : \mathcal{O} \rightarrow \mathbb{R}^m$ be locally Lipschitz continuous at $x \in \mathcal{O}$. The B-subdifferential of F at x is defined by*

$$\partial_B F(x) := \left\{ \lim_{k \rightarrow \infty} F'(x^k) | x^k \in D_F, x^k \rightarrow x \right\}.$$

The set $\partial F(x) = \text{co}(\partial_B F(x))$ is called Clarke's generalized Jacobian, where co denotes the convex hull.

The notion of semi-smoothness plays a key role on establishing locally superlinear convergence of the nonsmooth Newton-type method. Semi-smoothness was originally introduced by Mifflin [23] for real-valued functions and extended to vector-valued mappings by Qi and Sun [30].

DEFINITION 3.2. Let $F : \mathcal{O} \rightarrow \mathbb{R}^m$ be a locally Lipschitz continuous function. We say that F is semi-smooth at $x \in \mathcal{O}$ if

- (a) F is directionally differentiable at x ; and
- (b) for any $d \in \mathcal{O}$ and $J \in \partial F(x + d)$,

$$\|F(x + d) - F(x) - Jd\|_2 = o(\|d\|_2) \quad \text{as } d \rightarrow 0.$$

Furthermore, F is said to be strongly semi-smooth at $x \in \mathcal{O}$ if F is semi-smooth and for any $d \in \mathcal{O}$ and $J \in \partial F(x + d)$,

$$\|F(x + d) - F(x) - Jd\|_2 = O(\|d\|_2^2) \quad \text{as } d \rightarrow 0.$$

(Strongly) semi-smoothness is closed under scalar multiplication, summation and composition. The examples of semi-smooth functions include the smooth functions, all convex functions (thus norm), and the piecewise differentiable functions. Differentiable functions with Lipschitz gradients are strongly semi-smooth. For every $p \in [1, \infty]$, the norm $\|\cdot\|_p$ is strongly semi-smooth. Piecewise affine functions are strongly semi-smooth, such as $[x]_+ = \max\{0, x\}$. A vector-valued function is (strongly) semi-smooth if and only if each of its component functions is (strongly) semi-smooth. Examples of semi-smooth functions are thoroughly studied in [12, 37].

The basic properties of proximal mapping is well documented in textbooks such as [31, 2]. The proximal mapping prox_f , corresponding to a proper, closed and convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, is single-valued, maximal monotone and nonexpansive. Moreover, the proximal mappings of many interesting functions are (strongly) semi-smooth. It is worth mentioning that the semi-smoothness of proximal mapping does not hold in general [33]. The following lemma is useful when the proximal mapping of a function is complicate but the proximal mapping of its conjugate is easy.

LEMMA 3.3 (Moreau's decomposition). Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a proper, closed and convex function. Then, for any $t > 0$ and $x \in \mathbb{R}^n$,

$$x = \text{prox}_{tf}(x) + t\text{prox}_{f^*/t}(x/t).$$

We next review some existing results on the semi-smoothness of proximal mappings of various interesting functions. The proximal mapping of ℓ_1 -norm $\|x\|_1$, which is the well-known soft-thresholding operator, is component-wise separable and piecewise affine. Hence, the operator $\text{prox}_{\|\cdot\|_1}$ is strongly semi-smooth. According to the Moreau's decomposition, the proximal mapping of ℓ_∞ norm (the conjugate of ℓ_1 norm) is also strongly semi-smooth. For $k \in \mathbb{N}$, a function with k continuous derivatives is called a \mathcal{C}^k function. A function $f : \mathcal{O} \rightarrow \mathbb{R}^m$ defined on the open set $\mathcal{O} \subseteq \mathbb{R}^n$ is called piecewise \mathcal{C}^k function, $k \in [1, \infty]$, if f is continuous and if at every point $\bar{x} \in \mathcal{O}$ there exists a neighborhood $V \subset \mathcal{O}$ and a finite collection of \mathcal{C}^k functions $f_i : V \rightarrow \mathbb{R}^m$, $i = 1, \dots, N$, such that

$$f(x) \in \{f_1(x), \dots, f_N(x)\} \quad \text{for all } x \in V.$$

For a comprehensive study on piecewise \mathcal{C}^k functions, the readers are referred to [32]. From [37, Proposition 2.26], if f is a piecewise \mathcal{C}^1 (piecewise smooth) function, then f is semi-smooth; if f is a piecewise \mathcal{C}^2 function, then f is strongly semi-smooth. As described in [28, Section 5], in many applications the proximal mappings are piecewise \mathcal{C}^1 and thus semi-smooth. Metric projection, which is the proximal mapping of an indicator function, plays an important role in the analysis of constrained programs. The projection over a polyhedral set is piecewise linear [31, Example 12.31] and hence strongly semi-smooth. The projections over symmetric cones are proved to be strongly semi-smooth in [35].

3.2. Monotonicity of fixed-point mappings. This subsection focuses on the discussion of the monotonicity of the fixed-point mapping $F := I - T$, where $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a fixed-point operator. Later, we will show that the monotone property of F plays a critical role in our proposed method.

For the sake of readability, let us first recall some related concepts. A mapping $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is called *strongly monotone* with modulus $c > 0$ if

$$\langle x - y, F(x) - F(y) \rangle \geq c \|x - y\|_2^2, \quad \text{for all } x, y \in \mathbb{R}^n.$$

It is said that F is *cocoercive* with modulus $\beta > 0$ if

$$\langle x - y, F(x) - F(y) \rangle \geq \beta \|F(x) - F(y)\|_2^2, \quad \text{for all } x, y \in \mathbb{R}^n.$$

We now present the monotone properties of the fixed-point mappings $F_{\text{FBS}} = I - T_{\text{FBS}}$ and $F_{\text{DRS}} = I - T_{\text{DRS}}$.

PROPOSITION 3.4.

- (i) Suppose that ∇h is cocoercive with $\beta > 0$, then F_{FBS} is monotone if $0 < t \leq 2\beta$.
- (ii) Suppose that ∇h is strongly monotone with $c > 0$ and Lipschitz with $L > 0$, then F_{FBS} is strongly monotone if $0 < t < 2c/L^2$.
- (iii) Suppose that $h \in \mathcal{C}^2$, $H(x) := \nabla^2 h(x)$ is positive semidefinite for any $x \in \mathbb{R}^n$ and $\bar{\lambda} = \max_x \lambda_{\max}(H(x)) < \infty$. Then, F_{FBS} is monotone if $0 < t \leq 2/\bar{\lambda}$.
- (iv) The fixed-point mapping $F_{\text{DRS}} := I - T_{\text{DRS}}$ is monotone.

Proof. Items (i) and (ii) are well known in the literature, see [43] for example.

(iii) From the mean value theorem, there exists some x' such that

$$\nabla h(x) - \nabla h(y) = H(x')(x - y).$$

Hence, $\|\nabla h(x) - \nabla h(y)\|^2 \leq \bar{\lambda} \langle x - y, \nabla h(x) - \nabla h(y) \rangle$, which implies that ∇h is cocoercive with $1/\bar{\lambda}$. Hence, the monotonicity is obtained from item (i).

(iv) It has been shown that the operator T_{DRS} is firmly nonexpansive, see [21]. Therefore, F_{DRS} is firmly nonexpansive and hence monotone [2, Proposition 4.2]. \square

Items (i) and (ii) demonstrate that F_{FBS} is monotone as long as the step size t is properly selected. It is also shown in [43] that, when f is an indicator function of a convex closed set, the step size interval in items (i) and (ii) can be enlarged to $(0, 4\beta]$ and $(0, 4c/L^2)$, respectively. Item (iii) can also be found in [17, Lemma 4.1]. Finally, we introduce an useful lemma on the positive semidefinite property of the subdifferential of the monotone mapping.

LEMMA 3.5. For a monotone and Lipschitz continuous mapping $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and any $x \in \mathbb{R}^n$, each element of $\partial_B F(x)$ is positive semidefinite.

Proof. We first show that $F'(\bar{x})$ is positive semidefinite at a differentiable point \bar{x} . Suppose that there exist constant $a > 0$ and $d \in \mathbb{R}^n$ with $\|d\|_2 = 1$ such that $\langle d, F'(\bar{x})d \rangle = -a$. For any $t > 0$, let $\Phi(t) := F(\bar{x} + td) - F(\bar{x}) - tF'(\bar{x})d$. Since F is differentiable at \bar{x} , we

have $\|\Phi(t)\|_2 = o(t)$ as $t \rightarrow 0$. The monotonicity of F indicates that

$$\begin{aligned} 0 &\leq \langle td, F(\bar{x} + td) - F(\bar{x}) \rangle = \langle td, tF'(\bar{x})d + \Phi(t) \rangle \\ &\leq -at^2 + t\|d\|_2\|\Phi(t)\|_2 = -at^2 + o(t^2), \end{aligned}$$

which leads to a contradictory.

For any $x \in \mathbb{R}^n$ and each $J \in \partial_B F(x)$, there exists a sequence of differentiable points $x^k \rightarrow x$ such that $F'(x^k) \rightarrow J$. Since every $F'(x^k)$ is positive semidefinite, we have that J is also positive semidefinite. \square

3.3. A Regularized semi-smooth Newton method with Projection steps. The system of monotone equations has various applications [26, 34, 45, 20, 1]. Inspired by a pioneer work [34], a class of iterative methods for solving nonlinear (smooth) monotone equations were proposed in recent years [45, 20, 1]. In [34], the authors proposed a globally convergent Newton method by exploiting the structure of monotonicity, whose primary advantage is that the whole sequence of the distances from the iterates to the solution set is decreasing. The method is extended in [44] to solve monotone equations without nonsingularity assumption.

The main concept in [34] is introduced as follows. For an iterate z^k , let d^k be a descent direction such that

$$\langle F(u^k), -d^k \rangle > 0,$$

where $u^k = z^k + d^k$ is an intermediate iterate. By monotonicity of F , for any $z^* \in Z^*$ one has

$$\langle F(u^k), z^* - u^k \rangle \leq 0.$$

Therefore, the hyperplane

$$H_k := \{z \in \mathbb{R}^n \mid \langle F(u^k), z - u^k \rangle = 0\}$$

strictly separates z^k from the solution set Z^* . Based on this fact, it was developed in [34] that the next iterate is set by

$$z^{k+1} = z^k - \frac{\langle F(u^k), z^k - u^k \rangle}{\|F(u^k)\|_2^2} F(u^k).$$

It is easy to show that the point z^{k+1} is the projection of z^k onto the hyperplane H_k . The hyperplane projection step is critical to construct a globally convergent method for solving the system of nonlinear monotone equations. By applying the same technique, we develop a globally convergent method for solving semi-smooth monotone equations (3.1).

It has been demonstrated in Lemma 3.5 that each element of the B-subdifferential of a monotone and semi-smooth mapping is positive semidefinite. Hence, for an iterate z^k , by choosing an element $J_k \in \partial_B F(z^k)$, it is natural to apply a regularized Newton method. It computes

$$(3.2) \quad (J_k + \mu_k I)d = -F^k,$$

where $F^k = F(z^k)$, $\mu_k = \lambda_k \|F^k\|_2$ and $\lambda_k > 0$ is a regularization parameter. The regularization term $\mu_k I$ is chosen such that $J_k + \mu_k I$ is invertible. From a computational view, it is practical to solve the linear system (3.2) inexactly. Define

$$(3.3) \quad r^k := (J_k + \mu_k I)d^k + F^k.$$

At each iteration, we seek a step d^k by solving (3.2) approximately such that

$$(3.4) \quad \|r^k\|_2 \leq \tau \min\{1, \lambda_k \|F^k\|_2 \|d^k\|_2\},$$

where $0 < \tau < 1$ is some positive constant. Then a trial point is obtained as

$$u^k = z^k + d^k.$$

Define a ratio

$$(3.5) \quad \rho_k = \frac{-\langle F(u^k), d^k \rangle}{\|d^k\|_2^2}.$$

Select some parameters $0 < \eta_1 \leq \eta_2 < 1$ and $1 < \gamma_1 < \gamma_2$. If $\rho_k \geq \eta_1$, the iteration is said to be *successful*. Otherwise, the iteration is *unsuccessful*. Moreover, for a successful iteration, if $\|F(u^k)\|_2$ is sufficiently decreased, we take a Newton step, otherwise we take a hyperplane projection step. In summary, we set

$$(3.6) \quad z^{k+1} = \begin{cases} u^k, & \text{if } \rho_k \geq \eta_1 \text{ and } \|F(u^k)\|_2 \leq \nu \|F(\bar{u})\|_2, & \text{[Newton step]} \\ v^k, & \text{if } \rho_k \geq \eta_1 \text{ and } \|F(u^k)\|_2 > \nu \|F(\bar{u})\|_2, & \text{[projection step]} \\ z^k, & \text{otherwise.} & \text{[unsuccessful iteration]} \end{cases}$$

where $0 < \nu < 1$,

$$(3.7) \quad v^k = z^k - \frac{\langle F(u^k), z^k - u^k \rangle}{\|F(u^k)\|_2^2} F(u^k),$$

and the reference point \bar{u} is the iteration from the last Newton step. More specifically, when $\rho_k \geq \eta_1$ and $\|F(u^k)\|_2 \leq \nu \|F(\bar{u})\|_2$, we take $z^{k+1} = u^k$ and update $\bar{u} = u^k$.

Finally, the regularization parameter λ_k is updated as

$$(3.8) \quad \lambda_{k+1} \in \begin{cases} (\underline{\lambda}, \lambda_k), & \text{if } \rho_k \geq \eta_2, \\ [\lambda_k, \gamma_1 \lambda_k], & \text{if } \eta_1 \leq \rho_k < \eta_2, \\ (\gamma_1 \lambda_k, \gamma_2 \lambda_k], & \text{otherwise,} \end{cases}$$

where $\underline{\lambda} > 0$ is a small positive constant. These parameters determine how aggressively the regularization parameter is decreased when an iteration is successful or it is increased when an iteration is unsuccessful. The complete approach to solve (3.1) is summarized in Algorithm 1.

3.4. Global convergence. It is clear that a solution is obtained if Algorithm 1 terminates in finitely many iterations. Therefore, we assume that Algorithm 1 always generates an infinite sequence $\{z^k\}$ and $d^k \neq 0$ for any $k \geq 0$. Let Z^* be the solution set of system (3.1). Throughout this section, we assume that Z^* is nonempty. The following assumption is used in the sequel.

ASSUMPTION 3.6. Assume that $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is strongly semi-smooth and monotone. Suppose that there exists a constant $c_1 > 0$ such that $\|J_k\| \leq c_1$ for any $k \geq 0$ and any $J_k \in \partial_B F(z^k)$.

The following lemma demonstrates that the distance from z^k to Z^* decreases in a projection step. The proof follows directly from [34, Lemma 2.1], and it is omitted.

LEMMA 3.7. For any $z^* \in Z^*$ and any projection step, indexed by say k , we have that

$$(3.9) \quad \|z^{k+1} - z^*\|_2^2 \leq \|z^k - z^*\|_2^2 - \|z^{k+1} - z^k\|_2^2.$$

Algorithm 1: *An Adaptive Semi-smooth Newton (ASSN) method*

```

1 Give  $0 < \tau, \nu < 1$ ,  $0 < \eta_1 \leq \eta_2 < 1$  and  $1 < \gamma_1 \leq \gamma_2$ ;
2 Choose  $z^0$  and  $\varepsilon > 0$ . Set  $k = 0$  and  $\bar{u} = z^0$ ;
3 while not “converged” do
4   Select  $J_k \in \partial_B F(x^k)$ ;
5   Solve the linear system (3.2) approximately such that  $d^k$  satisfies (3.4);
6   Compute  $u^k = z^k + d^k$  and calculate the ratio  $\rho_k$  as in (3.5);
7   Update  $z^{k+1}$  and  $\lambda_{k+1}$  according to (3.6) and (3.8), respectively;
8   Set  $k = k + 1$ ;

```

Recall that F is strongly semi-smooth. Then for a point $z \in \mathbb{R}^n$ there exists $c_2 > 0$ (dependent on z) such that for any $d \in \mathbb{R}^n$ and any $J \in \partial_B F(z + d)$,

$$(3.10) \quad \|F(z + d) - F(z) - Jd\|_2 \leq c_2 \|d\|_2^2, \quad \text{as } \|d\|_2 \rightarrow 0.$$

Denote the index sets of Newton steps, projection steps and successful iterations, respectively, by

$$\begin{aligned} \mathcal{K}_N &:= \{k \geq 0 : \rho_k \geq \eta_1, \|F(u^k)\| \leq \nu \|F(\bar{u})\|\}, \\ \mathcal{K}_P &:= \{k \geq 0 : \rho_k \geq \eta_1, \|F(u^k)\| > \nu \|F(\bar{u})\|\} \end{aligned}$$

and

$$\mathcal{K}_S := \{k \geq 0 : \rho_k \geq \eta_1\}.$$

We next show that if there are only finitely many successful iterations, the later iterates are optimal solutions.

LEMMA 3.8. *Suppose that Assumption 3.6 holds and the index set \mathcal{K}_S is finite. Then $z^k = z^*$ for all sufficiently large k and $F(z^*) = 0$.*

Proof. Denote the index of the last successful iteration by k_0 . The construction of the algorithm implies that $z^{k_0+i} = z^{k_0+1} := z^*$, for all $i \geq 1$ and additionally $\lambda_k \rightarrow \infty$. Suppose that $a := \|F(z^*)\|_2 > 0$. For all $k > k_0$, it follows from (3.3) that

$$d^k = (J_k + \lambda_k \|F^k\|_2 I)^{-1} (r^k - F^k),$$

which, together with $\lambda_k \rightarrow \infty$, $\|r^k\|_2 \leq \tau$ and the fact that J_k is positive semidefinite, imply that $d^k \rightarrow 0$, and hence $u^k \rightarrow z^*$.

We now show that when λ_k is large enough, the ratio ρ_k is not smaller than η_2 . For this purpose, we consider an iteration with index $k > k_0$ sufficiently large such that $\|d^k\|_2 \leq 1$ and

$$\lambda_k \geq \frac{\eta_2 + c_1 + c_2}{a - \tau a}.$$

Then, it yields that

$$\begin{aligned} -\langle F(z^k), d^k \rangle &= \langle (J_k + \lambda_k \|F^k\|_2 I) d^k, d^k \rangle - \langle r^k, d^k \rangle \\ (3.11) \quad &\geq \lambda_k \|F^k\|_2 \|d^k\|_2^2 - \tau \lambda_k \|F^k\|_2 \|d^k\|_2^2 \\ &\geq (\eta_2 + c_1 + c_2) \|d^k\|_2^2. \end{aligned}$$

Further, for any $J_{u^k} \in \partial_B F(u^k)$ we obtain

$$\begin{aligned}
& -\langle F(u^k), d^k \rangle \\
& = -\langle F(z^k), d^k \rangle - \langle J_{u^k} d^k, d^k \rangle + \langle -F(u^k) + F(z^k) + J_{u^k} d^k, d^k \rangle \\
& \geq -\langle F(z^k), d^k \rangle - c_1 \|d^k\|_2^2 - \|F(z^* + d^k) - F(z^*) - J_{u^k} d^k\|_2 \\
& \geq (\eta_2 + c_1 + c_2) \|d^k\|_2^2 - c_1 \|d^k\|_2^2 - c_2 \|d^k\|_2^2 \\
& = \eta_2 \|d^k\|_2^2,
\end{aligned}$$

where the first inequality is from Assumption 3.6 and the facts that $\|d^k\|_2 \leq 1$ and $z^k = z^*$, and the second inequality comes from (3.11) and (3.10). Hence, we have $\rho_k \geq \eta_2$, which generates a successful iteration and yields a contradiction. This completes the proof. \square

The following result shows that a solution is derived if the set \mathcal{K}_N is infinite.

LEMMA 3.9. *Let Assumption 3.6 hold. If the sequence $\{z^k\}$ contains infinitely many iterates resulting from Newton steps, i.e., $|\mathcal{K}_N| = \infty$, then $\{z^k\}$ converges to some point \bar{z} such that $F(\bar{z}) = 0$.*

Proof. We first show that a subsequence of $\{z^k\}$ converges to a solution of $F(z) = 0$. Let $(k_i)_{i \geq 0}$ enumerate all elements of the set $\{k + 1 : k \in \mathcal{K}_N\}$ in increasing order. Since $\|F(z^{k_i})\|_2 \leq \nu \|F(z^{k_{i-1}})\|_2$ and $0 < \nu < 1$, we have that the subsequence $\{z^{k_i}\}$ converges to a solution \bar{z} as $i \rightarrow \infty$.

For any $k \notin \mathcal{K}_N$, we have $\|z^{k+1} - \bar{z}\|_2 \leq \|z^k - \bar{z}\|_2$ from the updating rule (3.6) and Lemma 3.7. Moreover, for any $k \notin \mathcal{K}_N$, there exists an index i such that $k_i < k + 1 < k_{i+1}$, and hence $\|z^{k+1} - \bar{z}\|_2 \leq \|z^{k_i} - \bar{z}\|_2$. Therefore, the whole sequence $\{z^k\}$ converges to \bar{z} . \square

We are now ready to prove the main global convergence result. In specific, we show that the infinite sequence $\{z^k\}$ generated by Algorithm 1 always converges to some solution.

THEOREM 3.10. *Let Assumption 3.6 hold. Then $\{z^k\}$ converges to some point \bar{z} such that $F(\bar{z}) = 0$.*

Proof. If the index set \mathcal{K}_S is finite, the result is directly from Lemma 3.8. The case that \mathcal{K}_N is infinite has been established in Lemma 3.9. The remaining part of the proof is to deal with the occurrence of that \mathcal{K}_N is finite and \mathcal{K}_P is infinite. In this situation, without loss of generality, we can ignore \mathcal{K}_N and assume that $\mathcal{K}_S = \mathcal{K}_P$ in the sequel.

Let z^* be any point in solution set Z^* . By Lemma 3.7, for any $k \in \mathcal{K}_S$, it yields that

$$(3.12) \quad \|z^{k+1} - z^*\|_2^2 \leq \|z^k - z^*\|_2^2 - \|z^{k+1} - z^k\|_2^2.$$

Therefore, the sequence $\{\|z^k - z^*\|_2\}$ is non-increasing and convergent, the sequence $\{z^k\}$ is bounded, and

$$(3.13) \quad \lim_{k \rightarrow \infty} \|z^{k+1} - z^k\|_2 = 0.$$

By (3.3) and (3.4), it follows that

$$\|F^k\|_2 \geq \|(J_k + \lambda_k \|F^k\|_2 I) d^k\|_2 - \|r^k\|_2 \geq (1 - \tau) \lambda_k \|F^k\|_2 \|d^k\|_2,$$

which implies that $\|d^k\|_2 \leq 1/[(1 - \tau)\lambda]$. This inequality shows that $\{d^k\}$ is bounded, and $\{u^k\}$ is also bounded. By using the continuity of F , there exists a constant $c_3 > 0$ such that

$$\|F(u^k)\|_2^{-1} \geq c_3, \quad \text{for any } k \geq 0.$$

Using (3.6), for any $k \in \mathcal{K}_S$, we obtain that

$$\|z^{k+1} - z^k\|_2 = \frac{-\langle F(u^k), d^k \rangle}{\|F(u^k)\|_2} \geq c_3 \rho_k \|d^k\|_2^2,$$

which, together with (3.13), imply that

$$(3.14) \quad \lim_{k \rightarrow \infty, k \in \mathcal{K}_S} \rho_k \|d^k\|_2^2 = 0.$$

We next consider two possible cases:

$$\liminf_{k \rightarrow \infty} \|F^k\|_2 = 0 \quad \text{and} \quad \liminf_{k \rightarrow \infty} \|F^k\|_2 = c_4 > 0.$$

In the first case, the continuity of F and the boundedness of $\{z^k\}$ imply that the sequence $\{z^k\}$ has some accumulation point \hat{z} such that $F(\hat{z}) = 0$. Since z^* is an arbitrary point in Z^* , we can choose $z^* = \hat{z}$ in (3.12). Then $\{z^k\}$ converges to \hat{z} .

In the second case, by using the continuity of F and the boundedness of $\{z^k\}$ again, there exist constants $c_5 > c_6 > 0$ such that

$$c_6 \leq \|F^k\|_2 \leq c_5, \quad \text{for all } k \geq 0.$$

If λ_k is large enough such that $\|d^k\|_2 \leq 1$ and

$$\lambda_k \geq \frac{\eta_2 + c_1 + c_2}{(1 - \tau)c_6},$$

then by a similar proof as in Lemma 3.8 we have that $\rho_k \geq \eta_2$ and consequently $\lambda_{k+1} < \lambda_k$. Hence, it turns out that $\{\lambda_k\}$ is bounded from above, by say $\bar{\lambda} > 0$. Using (3.3), (3.4), Assumption 3.6 and the upper bound of $\{\lambda_k\}$, we have

$$\|F^k\|_2 \leq \|(J_k + \lambda_k \|F^k\|_2 I) d^k\|_2 + \|r^k\|_2 \leq (c_1 + (1 + \tau)c_5 \bar{\lambda}) \|d^k\|_2.$$

Hence, it follows that

$$\liminf_{k \rightarrow \infty} \|d^k\|_2 > 0.$$

Then, by (3.14), it must hold that

$$\lim_{k \rightarrow \infty, k \in \mathcal{K}_S} \rho_k = 0,$$

which yields a contradiction to the definition of \mathcal{K}_S . Hence the second case is not possible. The proof is completed. \square

As is already shown, the global convergence of our Algorithm is essentially guaranteed by the projection step. However, by noticing that (3.7) is in the form of $v^k = z^k - \alpha_k F(u^k)$ with $\alpha_k = \langle F(u^k), z^k - u^k \rangle / \|F(u^k)\|_2^2 > 0$, the projection step is indeed an extragradient step [12]. Since the asymptotic convergence rate of the extragradient step is often not faster than that of the Newton step, a slow convergence may be observed if the projection step is always performed. Hence, our modification (3.6) is practically meaningful. Moreover, we will next prove that the projection step will never be performed when the iterate is close enough to a solution under some generalized nonsingular conditions.

3.5. Fast local convergence. Since Algorithm 1 has been shown to be globally convergent, we now assume that the sequence $\{z^k\}$ generated by Algorithm 1 converges to a solution $z^* \in Z^*$. Under some reasonable conditions, we will prove that the Newton steps achieve a locally quadratic convergence. Moreover, we will show that when the iteration point z^k is close enough to z^* , the condition $\|F(u^k)\|_2 \leq \nu \|F(z^k)\|_2$ is always satisfied.

Consequently, Algorithm 1 turns into a second-order Newton method in a neighborhood of z^* .

We make the following assumption.

ASSUMPTION 3.11. *The mapping F is BD-regular at z^* , that is, all elements in $\partial_B F(z^*)$ are nonsingular.*

The BD-regularity is a common assumption in the analysis of the local convergence of nonsmooth methods. The following properties of the BD-regularity are directly derived from [29, Proposition 2.5] and [27, Proposition 3].

LEMMA 3.12. *Suppose that F is BD-regular at z^* , then there exist constants $c_0 > 0$, $\kappa > 0$ and a neighborhood $N(z^*, \varepsilon_0)$ such that for any $y \in N(z^*, \varepsilon_0)$ and $J \in \partial_B F(y)$,*

- (i) *J is nonsingular and $\|J^{-1}\| \leq c_0$;*
- (ii) *z^* is an isolated solution;*
- (iii) *the error bound condition holds for $F(z)$ and $N(z^*, \varepsilon_0)$, that is $\|y - z^*\|_2 \leq \kappa \|F(y)\|_2$.*

Since z^* is isolated, the term $\text{dist}(y, Z^*)$ in Definition 2.2 is degenerated to $\|y - z^*\|_2$ and it becomes the error bound condition in item (iii). The local convergence relies on some auxiliary results.

LEMMA 3.13. *Suppose that Assumption 3.11 holds true, then*

- (i) *the parameter λ_k is bounded above by some constant $\bar{\lambda} > 0$;*
- (ii) *there exists some $L > 0$ such that $\|F(z)\|_2 \leq L\|z - z^*\|_2$ for any $z \in N(z^*, \varepsilon_0)$;*
- (iii) *for any $z^k \in N(z^*, \varepsilon_1)$ with $\varepsilon_1 := \min\{\varepsilon_0, 1/(2Lc_0\tau\bar{\lambda})\}$, we have*

$$\|d^k\|_2 \leq 2c_0L\|z^k - z^*\|_2.$$

Proof. Item (i) has been shown in the proof of global convergence. The local Lipschitz continuity in item (ii) is obvious since F is semi-smooth. For any $z^k \in N(z^*, \varepsilon_1)$, one has $\|F^k\|_2 \leq L\|z^k - z^*\|_2 \leq L\varepsilon_1$, hence

$$(3.15) \quad c_0\tau\lambda_k\|F^k\|_2 \leq c_0\tau\bar{\lambda}\|F^k\|_2 \leq 1/2.$$

Note that

$$\begin{aligned} \|d^k\|_2 &\leq \|(J_k + \mu_k I)^{-1} F^k\|_2 + \|(J_k + \mu_k I)^{-1} r^k\|_2 \\ &\leq c_0L\|z^k - z^*\|_2 + c_0\tau\lambda_k\|F^k\|_2\|d^k\|_2, \end{aligned}$$

we have $(1 - c_0\tau\lambda_k\|F^k\|_2)\|d^k\|_2 \leq c_0L\|z^k - z^*\|_2$, which, together with (3.15), yields $\|d^k\|_2 \leq 2c_0L\|z^k - z^*\|_2$. \square

We next show that the Newton steps are locally quadratically convergent.

THEOREM 3.14. *Suppose that Assumption 3.11 holds. Then for any $k \in S_N$ and $z^k \in N(z^*, \varepsilon_1)$, we have*

$$(3.16) \quad \|z^{k+1} - z^*\|_2 \leq c_7\|z^k - z^*\|_2^2,$$

where the constant $c_7 := c_0(c_2 + (1 + 2c_0L\tau)\bar{\lambda}L)$.

Proof. For a Newton step, we have

$$\begin{aligned}
\|z^{k+1} - z^*\|_2 &= \|z^k + d^k - z^*\|_2 \\
&= \|z^k + (J_k + \mu_k I)^{-1}(F^k + (J_k + \mu_k I)d^k - F^k) - z^*\|_2 \\
&\leq \|z^k - z^* - (J_k + \mu_k I)^{-1}F^k\|_2 + \|(J_k + \mu_k I)^{-1}\| \cdot \|F^k + (J_k + \mu_k I)d^k\|_2 \\
&\leq \|(J_k + \mu_k I)^{-1}\| \cdot (\|F^k - F(z^*) - J_k(z^k - z^*)\|_2 + \mu_k \|z^k - z^*\|_2 + \|r^k\|_2) \\
&\leq \|J_k^{-1}\| \cdot (\|F^k - F(z^*) - J_k(z^k - z^*)\|_2 + \lambda_k \|F^k\|_2 \|z^k - z^*\|_2 + \tau \lambda_k \|F^k\|_2 \|d^k\|_2) \\
&\leq \|J_k^{-1}\| \cdot (c_2 \|z^k - z^*\|_2^2 + \lambda_k \|F^k\|_2 \|z^k - z^*\|_2 + \tau \lambda_k \|F^k\|_2 \|d^k\|_2) \\
&\leq c_0(c_2 \|z^k - z^*\|_2^2 + (1 + 2c_0 L \tau) \lambda_k \|F^k\|_2 \|z^k - z^*\|_2) \\
&\leq c_0(c_2 + (1 + 2c_0 L \tau) \bar{\lambda} L) \|z^k - z^*\|_2^2,
\end{aligned}$$

where the third inequality is from the facts that $\mu_k = \lambda_k \|F^k\|_2$ and $\|r^k\|_2 \leq \tau \lambda_k \|F^k\|_2 \|d^k\|_2$, the fourth inequality uses (3.10), and the fifth inequality arises from item (iii) in Lemma 3.13.

□

Based on Theorem 3.14, a region is defined in the following corollary. It is shown that, $\|F(u^k)\|_2 \leq \nu \|F^k\|_2$ is always satisfied in this region.

COROLLARY 3.15. *Under the conditions of Theorem 3.14, for any $z^k \in N(z^*, \varepsilon_2)$ with $\varepsilon_2 := \min\{\varepsilon_1, \nu/(Lc_7\kappa)\}$, we have $\|F(u^k)\|_2 \leq \nu \|F(z^k)\|_2$.*

Proof. Using the Lipschitz of F , Theorem 3.14 and item (iii) in Lemma 3.12, we obtain

$$\|F(u^k)\|_2 \leq L \|z^k + d^k - z^*\|_2 \leq Lc_7 \|z^k - z^*\|_2^2 \leq Lc_7 \varepsilon_2 \|z^k - z^*\|_2 \leq \nu \|F(z^k)\|_2.$$

□

It is clear that the BD-regular condition plays a key role in the above discussion. Although the BD-regular condition is strong and may fail in some situations, there are some possible ways to resolve this issue. As is shown in [24, Section 4.2], suppose that there exists a nonsingular element in $\partial_B F(z^*)$ and other elements in $\partial_B F(z^*)$ may be singular. By exploiting the structure of $\partial_B F(z)$, one can carefully choose a nonsingular generalized Jacobian when z is close enough to z^* . Hence, if z^* is isolated, one can still obtain the fast local convergence results by a similar proof as above. Another way is inspired by the literature on the Levenberg-Marquardt (LM) method. The LM method is a regularized Gauss-Newton method to deal with some possibly singular systems. It has been shown in [13] that the LM method preserves a superlinear or quadratic local convergence rate under certain local error bound condition, which is weaker than the nonsingular condition. Therefore, it remains a future research topic to investigate local convergence of our algorithm under the local error bound condition.

3.6. Regularized L-BFGS method with projection steps. In this subsection, we propose a regularized L-BFGS method with projection steps by simply replace the Newton step in Algorithm 1 with a regularized L-BFGS step to avoid solving the linear system (3.2). The L-BFGS method is an adaption of the classical BFGS method, which tries to use a minimal storage. A globally convergent BFGS method with projection steps is proposed in [45] for solving smooth monotone equations. The convergence of our regularized L-BFGS method can be analyzed in a similar way as our regularized Newton method by combining the convergence analysis in [45]. We only describe the L-BFGS update in the following and omit the convergence analysis.

For an iterate z^k , we compute the direction by

$$(3.17) \quad (H_k + \mu_k I)d^k = -F^k,$$

where H_k is the L-BFGS approximation to the Jacobian matrix.

Choosing an initial matrix H_k^0 and setting $\delta F^k = F^{k+1} - F^k$, the Jacobian matrix can be approximated by the recent m pairs $\{\delta F^i, d^i\}$, $i = k-1, k-2, \dots, k-m$, i.e., using the standard formula [25] as

$$(3.18) \quad H_k = H_k^0 - [H_k^0 \mathcal{D}_k \quad \mathcal{F}_k] \begin{bmatrix} \mathcal{D}_k^T H_k^0 \mathcal{D}_k & L_k \\ L_k^T & -S_k \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{D}_k^T (H_k^0)^T \\ \mathcal{F}_k^T \end{bmatrix},$$

where $\mathcal{D}_k = [d^{k-m}, \dots, d^{k-1}]$, $\mathcal{F}_k = [\delta F^{k-m}, \dots, \delta F^{k-1}]$, L_k is a lower-triangular matrix with entries

$$(L_k)_{i,j} = \begin{cases} (d^{k-m-1+i})^T (\delta F^{k-m-1+j}) & \text{if } i > j, \\ 0 & \text{otherwise,} \end{cases}$$

and S_k is a diagonal matrix with entries

$$(S_k)_{ii} = (d^{k-m-1+i})^T \delta F^{k-m-1+i}.$$

Then we can compute the inverse regularized Jacobian matrix

$$(H_k + \mu_k I)^{-1} = \bar{H}_k^{-1} + \bar{H}_k^{-1} C_k R_k^{-1} C_k^T (\bar{H}_k^T)^{-1},$$

where $\bar{H}_k = H_k^0 + \mu_k I$, $C_k = [H_k^0 \mathcal{D}_k \quad \mathcal{F}_k]$, R_k is defined by $R_k = V_k - C_k^T \bar{H}_k^{-1} C_k$ and

$$V_k = \begin{bmatrix} \mathcal{D}_k^T H_k^0 \mathcal{D}_k & L_k \\ L_k^T & -S_k \end{bmatrix}.$$

Specifically, if k is smaller than m , we use the classical BFGS method to approximate inverse regularized Jacobian matrix, which just let $d^j = \delta F^j = 0$ for $j < 0$ in the formula (3.18).

4. Numerical Results. In this section, we conduct proof-of-concept numerical experiments on our proposed schemes for the fixed-point mappings induced from the FBS and DRS methods by applying them to ℓ_1 -norm minimization problem. All numerical experiments are performed in MATLAB on workstation with a Intel(R) Xeon(R) CPU E5-2680 v3 and 128GB memory.

4.1. Applications to the FBS method. Consider the ℓ_1 -regularized optimization problem of the form

$$(4.1) \quad \min \mu \|x\|_1 + h(x),$$

where h is continuously differentiable. Let $f(x) = \mu \|x\|_1$. The system of nonlinear equations corresponding to the FBS method is $F(x) = x - \text{prox}_{t f}(x - t \nabla h(x)) = 0$. The generalized Jacobian matrix of $F(x)$ is

$$(4.2) \quad J(x) = I - M(x)(I - t \partial^2 h(x)),$$

where $M(x) \in \partial \text{prox}_{t f}(x - t \nabla h(x))$ and $\partial^2 h(x)$ is the generalized Hessian matrix of $h(x)$. Specifically, the proximal mapping corresponding to $f(x)$ is the so-called shrinkage operator defined as

$$(\text{prox}_{t f}(x))_i = \text{sign}(x_i) \max(|x_i| - \mu t, 0).$$

Hence, one can take a Jacobian matrix $M(x)$ which is a diagonal matrix with diagonal entries being

$$(M(x))_{ii} = \begin{cases} 1, & \text{if } |(x - t\nabla h(x))_i| > \mu t, \\ 0, & \text{otherwise.} \end{cases}$$

Similar to [24], we introduce the index sets

$$\begin{aligned} \mathcal{I}(x) &:= \{i : |(x - t\nabla h(x))_i| > t\mu\} = \{i : (M(x))_{ii} = 1\}, \\ \mathcal{O}(x) &:= \{i : |(x - t\nabla h(x))_i| \leq t\mu\} = \{i : (M(x))_{ii} = 0\}. \end{aligned}$$

The Jacobian matrix can be represented by

$$J(x) = \begin{pmatrix} t(\partial^2 h(x))_{\mathcal{I}(x)\mathcal{I}(x)} & t(\partial^2 h(x))_{\mathcal{I}(x)\mathcal{O}(x)} \\ 0 & I \end{pmatrix}.$$

Using the above special structure of Jacobian matrix $J(x)$, we can reduce the complexity of the regularized Newton step (3.2). Let $\mathcal{I} = \mathcal{I}(x^k)$ and $\mathcal{O} = \mathcal{O}(x^k)$. Then, we have

$$\begin{aligned} (1 + \mu_k)s_{\mathcal{O}}^k &= -F_{k,\mathcal{O}}, \\ (t(\partial^2 h(x))_{\mathcal{I}\mathcal{I}} + \mu I)s_{\mathcal{I}}^k + t(\partial^2 h(x))_{\mathcal{I}\mathcal{O}}s_{\mathcal{O}}^k &= -F_{k,\mathcal{I}}, \end{aligned}$$

which yields

$$\begin{aligned} s_{\mathcal{O}}^k &= -\frac{1}{1 + \mu_k}F_{k,\mathcal{O}}, \\ (t(\partial^2 h(x))_{\mathcal{I}\mathcal{I}} + \mu I)s_{\mathcal{I}}^k &= -F_{k,\mathcal{I}} - t(\partial^2 h(x))_{\mathcal{I}\mathcal{O}}s_{\mathcal{O}}^k. \end{aligned}$$

4.1.1. Numerical comparison. In this subsection, we compare our proposed methods with different solvers for solving problem (4.1) with $h(x) = \frac{1}{2}\|Ax - b\|_2^2$. The solvers used for comparison include ASSN, SSNP, ALSB, FPC-AS [39], SpaRSA [40] and SNF [24]. ASSN is the proposed semi-smooth Newton method with projection steps (Algorithm 1) and SSNP is the method which only uses the projection steps. ASLB(i) is a variant of the line search based method by combining the L-BFGS method and hyperplane projection technique. The number in bracket is the size of memory. FPC-AS is a first-order method that uses a fixed-point iteration under Barzilai-Borwein steps and continuation strategy. SpaRSA resembles FPC-AS, which is also a first-order methods and uses Barzilai-Borwein steps and continuation strategy. SNF is a semi-smooth Newton type method which uses the filter strategy and is one of state-of-the-art second-order methods for ℓ_1 -regularized optimization problem (4.1) and SNF(aCG) is the SNF solver with an adaptive parameter strategy in the conjugate gradient method. The parameters of FPC-AS, SpaRSA and SNF are the same as [24].

The test problems are from [24], which are constructed as follows. Firstly, we randomly generate a sparse solution $\bar{x} \in \mathbb{R}^n$ with k nonzero entries, where $n = 512^2 = 262144$ and $k = \lceil n/40 \rceil = 5553$. The k different indices are uniformly chosen from $\{1, 2, \dots, n\}$ and we set the magnitude of each nonzero element by $\bar{x}_i = \eta_1(i)10^{d\eta_2(i)/20}$, where $\eta_1(i)$ is randomly chosen from $\{-1, 1\}$ with probability 1/2, respectively, $\eta_2(i)$ is uniformly distributed in $[0, 1]$ and d is a dynamic range which can influence the efficiency of the solvers. Then we choose $m = n/8 = 32768$ random cosine measurements, i.e., $Ax = (\text{dct}(x))_J$, where J contains m different indices randomly chosen from $\{1, 2, \dots, n\}$ and dct is the discrete cosine transform. Finally, the input data is specified by $b = A\bar{x} + \epsilon$, where ϵ is a Gaussian noise with a standard deviation $\bar{\sigma} = 0.1$.

To compare fairly, we set an uniform stopping criterion. For a certain tolerance ϵ , we obtain a solution x_{newt} using ASSN such that $\|F(x_{newt})\| \leq \epsilon$. Then we terminate all methods by the relative criterion

$$\frac{f(x^k) - f(x^*)}{\max\{f(x^*), 1\}} \leq \frac{f(x_{newt}) - f(x^*)}{\max\{f(x^*), 1\}},$$

where $f(x)$ is the objective function and x^* is a highly accurate solution obtained by ASSN under the criterion $\|F(x)\| \leq 10^{-14}$.

TABLE 4.1

Total number of A - and A^T - calls N_A and CPU time (in seconds) averaged over 10 independent runs with dynamic range 20 dB

method	$\epsilon : 10^0$		$\epsilon : 10^{-1}$		$\epsilon : 10^{-2}$		$\epsilon : 10^{-4}$		$\epsilon : 10^{-6}$	
	time	N_A	time	N_A	time	N_A	time	N_A	time	N_A
SNF	1.12	84.6	2.62	205	3.19	254.2	3.87	307	4.5	351
SNF(aCG)	1.11	84.6	2.61	205	3.19	254.2	4.19	331.2	4.3	351.2
ASSN	1.15	89.8	1.81	145	2.2	173	3.15	246.4	3.76	298.2
SSNP	2.52	199	5.68	455.6	8.05	649.4	20.7	1679.8	29.2	2369.6
ASLB(2)	0.803	57	1.35	98.4	1.66	121	2.79	202.4	3.63	264.6
ASLB(1)	0.586	42.2	1.01	71.6	1.29	92	2.54	181.4	3.85	275
FPC-AS	1.45	109.8	5.03	366	7.08	510.4	10	719.8	10.3	743.6
SpaRSA	5.46	517.2	5.54	519.2	5.9	539.8	6.75	627	9.05	844.4

TABLE 4.2

Total number of A - and A^T - calls N_A and CPU time (in seconds) averaged over 10 independent runs with dynamic range 40 dB

method	$\epsilon : 10^0$		$\epsilon : 10^{-1}$		$\epsilon : 10^{-2}$		$\epsilon : 10^{-4}$		$\epsilon : 10^{-6}$	
	time	N_A	time	N_A	time	N_A	time	N_A	time	N_A
SNF	2.12	158.2	4.85	380.8	6.07	483.2	6.8	525	7.2	562.4
SNF(aCG)	2.07	158.2	4.84	380.8	6.1	483.2	7.1	553.6	7.22	573.6
ASSN	2.34	182.2	3.67	285.4	4.29	338.6	5.11	407	5.92	459.2
SSNP	6.05	485.6	12.3	978.6	19.5	1606.6	27.3	2190.8	37.1	2952.2
ASLB(2)	1.39	98.2	2.19	154.4	2.64	194	3.45	250.4	4.49	323.6
ASLB(1)	1.25	86.8	1.84	127.4	2.2	161.6	3.2	225.6	4.59	319.2
FPC-AS	2.08	158	5.31	399.4	7.8	578.6	10.1	720.4	10.5	775
SpaRSA	5.56	523.4	5.56	530	6.27	588.2	7.45	671.6	8.11	759.6

We solve the test problems under different tolerances $\epsilon \in \{10^{-0}, 10^{-1}, 10^{-2}, 10^{-4}, 10^{-6}\}$ and dynamic ranges $d \in \{20, 40, 60, 80\}$. Since the evaluations of dct dominate the overall computation, we mainly use the total numbers of A - and A^T - calls N_A to compare the efficiency of different solvers. Tables 4.1 - 4.4 show the averaged numbers of N_A and CPU time over 10 independent trials. These tables show that ASSN and ASLB are competitive to other methods. For the low accuracy, SpaRSA and FPC-AS show a fast convergence rate. ASSN and ASLB are both faster than or close to FPC-AS and SpaRSA regardless of N_A and CPU time in most cases. In the meanwhile, ASSN and ASLB are competitive to the second-order methods under moderate accuracy. The CPU time and N_A of ASSN and ASLB are less than the Newton type solver SNF in almost all cases, especially for the large dynamic range. ASLB with a memory size $m = 1$ shows the fastest speed in low accuracy. It is necessary

TABLE 4.3

Total number of A - and A^T - calls N_A and CPU time (in seconds) averaged over 10 independent runs with dynamic range 60 dB

method	$\epsilon : 10^0$		$\epsilon : 10^{-1}$		$\epsilon : 10^{-2}$		$\epsilon : 10^{-4}$		$\epsilon : 10^{-6}$	
	time	N_A	time	N_A	time	N_A	time	N_A	time	N_A
SNF	5.66	391.8	9.31	648.8	11.1	777.6	11.8	828.2	12.5	876.6
SNF(aCG)	5.62	391.8	9.28	648.8	11	777.6	12.2	861.2	12.7	889
ASSN	3.92	295.4	5.38	416.4	6.45	492	7.49	582.4	8.19	642.4
SSNP	21.5	1607.2	29.5	2247.6	32.2	2478.8	41.9	3236.2	50.9	3927.4
ASLB(2)	2.11	146.2	2.89	201.6	3.54	250.6	4.5	317.6	5.42	383.4
ASLB(1)	2.11	143.8	2.66	187.8	3.25	228.2	4.22	295	5.22	368.6
FPC-AS	3.02	232.2	8.84	644	11.5	844.2	13.8	1004.2	14.6	1031.8
SpaRSA	6.01	561.2	6.39	598.2	7.27	683.2	8.25	797.8	9.84	900.6

TABLE 4.4

Total number of A - and A^T - calls N_A and CPU time (in seconds) averaged over 10 independent runs with dynamic range 80 dB

method	$\epsilon : 10^0$		$\epsilon : 10^{-1}$		$\epsilon : 10^{-2}$		$\epsilon : 10^{-4}$		$\epsilon : 10^{-6}$	
	time	N_A	time	N_A	time	N_A	time	N_A	time	N_A
SNF	7.47	591	10.7	841.6	12.4	978.6	13	1024.8	13.6	1057.8
SNF(aCG)	7.56	591	10.6	841.6	12.4	978.6	13.2	1042.2	13.9	1099.4
ASSN	6.39	482.8	7.66	601	8.66	690.6	9.9	780.6	10.5	833.4
SSNP	36.1	2820.6	34.2	2767.2	42.7	3497	51.3	4201.4	56.6	4531.2
ASLB(2)	3.65	255.8	4.03	299.4	4.98	355.6	5.61	411.4	6.21	440
ASLB(1)	3.02	213.6	3.59	258	4.24	299.2	4.95	357.6	5.52	385.6
FPC-AS	4.16	321.2	8.18	611.4	10.7	788.4	12.1	886	12.1	900.8
SpaRSA	5.74	543.2	6.96	665.4	8.17	763.2	9.1	873.6	9.85	930.2

to emphasize that L-BFGS with $m = 1$ is equal to the Hestenes-Stiefel and Polak-Ribière conjugate gradient method with exact line search [25]. Compared with ASSN, SSNP has a slower convergent rate, which implies that our adaptive strategy on switching Newton and projection steps is helpful.

In particular, ASSN and ASLB have a better performance for high accuracy. Figures 4.1 and 4.2 illustrate the residual history with respect to the total number of A - and A^T - calls N_A and the total number of iterations. Since two first-order methods have a close performance and ASLB(1) performs better than ASLB(2), we omit the figure of FPC-AS and ASLB(2). These figures also show that ASSN and ASLB have a better performance than SNF and SNF(aCG) independent of dynamic ranges. In particular, quadratic convergence is observable from ASSN in these examples.

4.2. Applications to the DRS method.

Consider the Basis-Pursuit (BP) problem

$$(4.3) \quad \min \|x\|_1, \text{ subject to } Ax = b,$$

where $A \in \mathbb{R}^{m \times n}$ is of full row rank and $b \in \mathbb{R}^m$. Let $f(x) = 1_\Omega(Ax - b)$ and $h(x) = \|x\|_1$, where the set $\Omega = \{0\}$. The system of nonlinear equations corresponding to the DRS fixed-point mapping is

$$(4.4) \quad F(z) = \text{prox}_{th}(z) - \text{prox}_{tf}(2\text{prox}_{th}(z) - z) = 0.$$

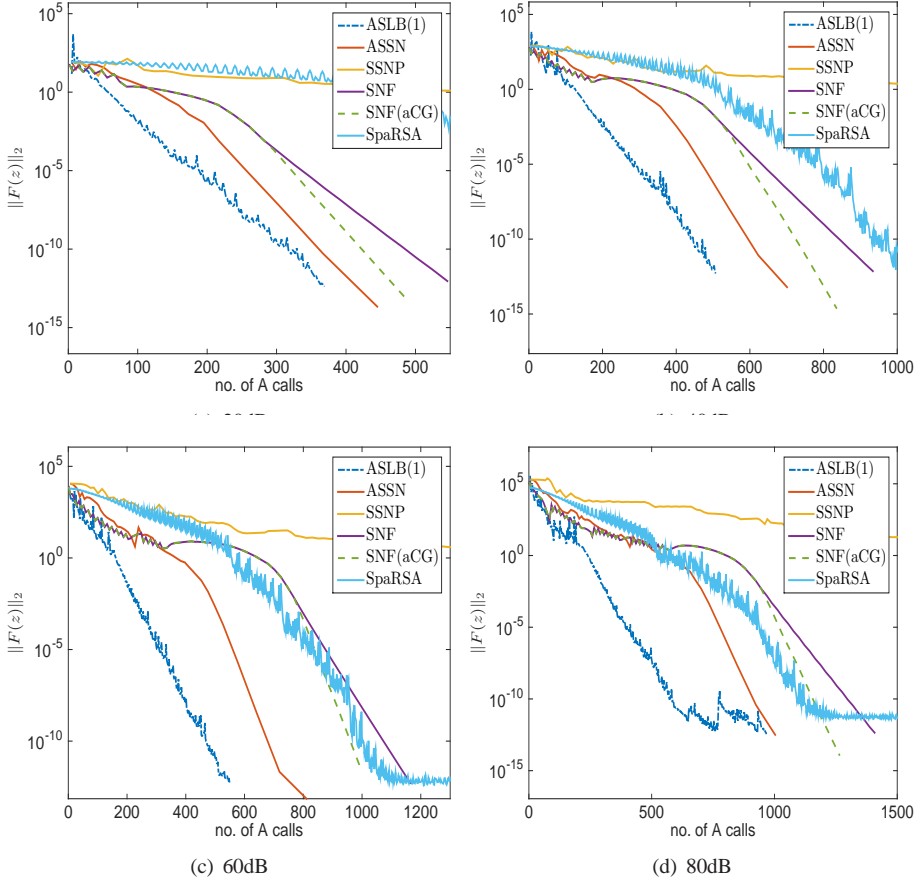


FIG. 4.1. residual history with respect to the total numbers of A - and A^T - calls N_A

For the simplicity of solving the subproblems in the DRS method, we make the assumption that $AA^T = I$. Then it can be derived that the proximal mapping with respect to $f(x)$ is

$$\begin{aligned} \mathbf{prox}_{tf}(z) &= (I - A^T A)z + A^T (\mathbf{prox}_{1_\Omega}(Az - b) + b) \\ &= z - A^T(Az - b). \end{aligned}$$

A generalized Jacobian matrix $D \in \partial \mathbf{prox}_{tf}((2\mathbf{prox}_{th}(z) - z))$ is taken as follows

$$(4.5) \quad D = I - A^T A.$$

The proximal mapping with respect to $h(x)$ is

$$(\mathbf{prox}_{th}(z))_i = \text{sign}(z_i) \max(|z_i| - t, 0).$$

One can take a generalized Jacobian matrix $M(z) \in \partial \mathbf{prox}_{th}(z)$ as a diagonal matrix with diagonal entries

$$M_{ii}(z) = \begin{cases} 1, & |z_i| > t, \\ 0, & \text{otherwise.} \end{cases}$$

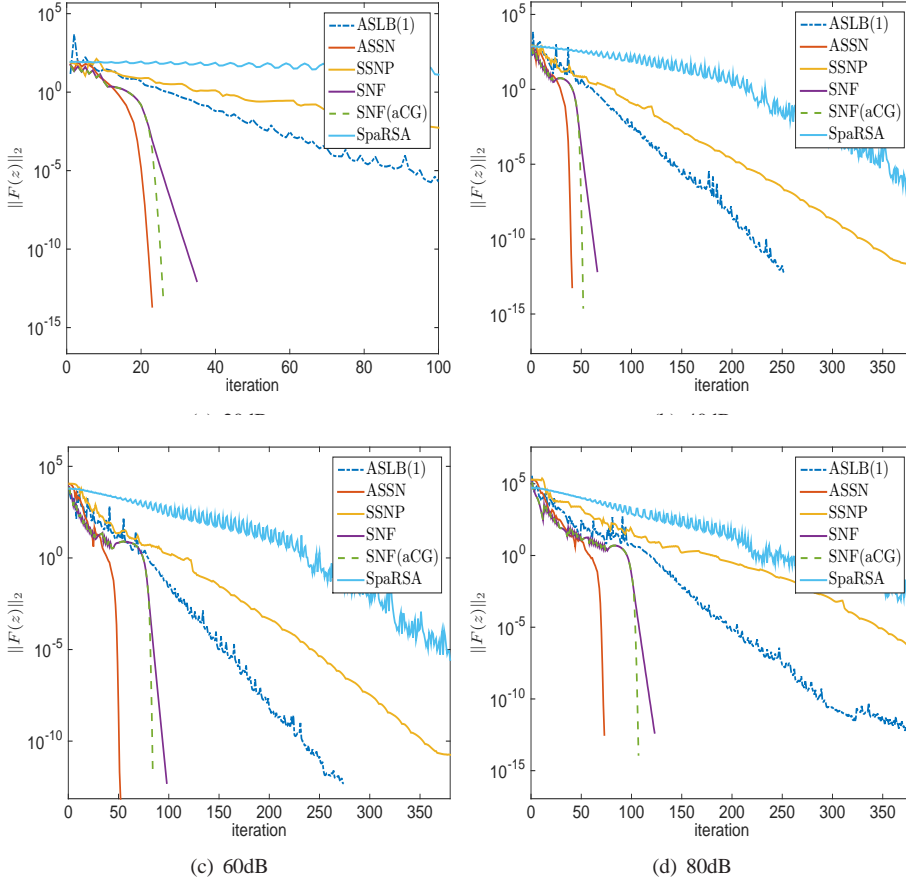


FIG. 4.2. residual history with respect to the total numbers of iteration

Hence, a generalized Jacobian matrix of $F(z)$ is in the form of

$$(4.6) \quad J(z) = M(z) + D(I - 2M(z)).$$

Let $W = (I - 2M(z))$ and $H = W + M(z) + \mu I$. Using the binomial inverse theorem, we obtain the inverse matrix

$$\begin{aligned} (J(z) + \mu I)^{-1} &= (H - A^\top A W)^{-1} \\ &= H^{-1} + H^{-1} A^\top (I - A W H^{-1} A^\top)^{-1} A W H^{-1}. \end{aligned}$$

For convenience, we write the diagonal entries of matrix W and H as

$$W_{ii}(z) = \begin{cases} -1, & |(z)_i| > t, \\ 1, & \text{otherwise} \end{cases} \quad \text{and} \quad H_{ii}(z) = \begin{cases} \mu, & |(z)_i| > t, \\ 1 + \mu, & \text{otherwise}. \end{cases}$$

Then $W H^{-1} = \frac{1}{1+\mu} I - S$, where S is a diagonal matrix with diagonal entries

$$S_{ii}(z) = \begin{cases} \frac{1}{\mu} + \frac{1}{1+\mu}, & |(z)_i| > t, \\ 0, & \text{otherwise}. \end{cases}$$

Hence, $I - AW H^{-1} A^\top = (1 - \frac{1}{1+\mu})I + ASA^\top$. Define the index sets

$$\begin{aligned}\mathcal{I}(x) &:= \{i : |(z)_i| > t\} = \{i : M_{ii}(x) = 1\}, \\ \mathcal{O}(x) &:= \{i : |(z)_i| \leq t\} = \{i : M_{ii}(x) = 0\}\end{aligned}$$

and $A_{\mathcal{I}(x)}$ denote the matrix containing the column $\mathcal{I}(x)$ of A , then we have

$$(4.7) \quad ASA^\top = (\frac{1}{\mu} + \frac{1}{1+\mu})A_{\mathcal{I}(x)}A_{\mathcal{I}(x)}^\top.$$

The above property implies the positive definiteness of $I - AW H^{-1} A^\top$ and can be used to reduce the computational complexity if the submatrix $A_{\mathcal{I}(x)}A_{\mathcal{I}(x)}^\top$ is easily available.

4.2.1. Numerical comparison. In this subsection, we compare our methods with two first-order solvers: ADMM [41] and SPGL1 [38]. The ASLB solver is not included since its performance is not comparable with other approaches. Our test problems are almost the same as the last subsection and the only difference is that we set $b = A\bar{x}$ without adding noise. We use the residual criterion $\|F(z)\| \leq \epsilon$ as the stopping criterion for ADMM and ASSN. Because the computation of residual of SPGL1 needs extra cost, we use its original criterion and list the relative error “rerr” to compare with ADMM and ASSN. The relative error with respect to the true solution x^* is denoted by

$$\text{rerr} = \frac{\|x^k - x^*\|}{\max(\|x^*\|, 1)}.$$

We revise the ADMM in `yall1`¹ by adjusting the rules of updating the penalty parameter and choosing the best parameters so that it can solve all examples in our numerical experiments. The parameters are set to the default values in SPGL1. Since the matrix A is only available as an operator, the property (4.7) cannot be applied in ASSN.

TABLE 4.5
Total number of A - and A^T -calls N_A , CPU time (in seconds) and relative error with dynamic range 20 dB

method	$\epsilon : 10^{-2}$			$\epsilon : 10^{-4}$			$\epsilon : 10^{-6}$		
	time	N_A	rerr	time	N_A	rerr	time	N_A	rerr
ADMM	10.9	646	2.781e-04	14	1026	2.658e-06	19.4	1438	2.467e-08
ASSN	8.58	694	1.175e-04	9.73	734	2.811e-06	10.7	813	4.282e-09
SPGL1	17.3	733	2.127e-01	54.4	2343	2.125e-01	72.3	3232	2.125e-01

TABLE 4.6
Total number of A - and A^T -calls N_A , CPU time (in seconds) and relative error with dynamic range 40 dB

method	$\epsilon : 10^{-2}$			$\epsilon : 10^{-4}$			$\epsilon : 10^{-6}$		
	time	N_A	rerr	time	N_A	rerr	time	N_A	rerr
ADMM	6.92	504	2.092e-04	12	875	2.623e-06	17.3	1306	2.926e-08
ASSN	5.79	469	7.595e-05	7.19	582	8.922e-07	8.43	632	2.006e-08
SPGL1	29.8	1282	2.350e-02	58.5	2477	2.346e-02	68.1	2910	2.346e-02

We solve the test problems under different tolerances $\epsilon \in \{10^{-2}, 10^{-4}, 10^{-6}\}$ and dynamic ranges $d \in \{20, 40, 60, 80\}$. Similar to the last subsection, we mainly use the total

¹downloadable from <http://yall1.blogs.rice.edu>

TABLE 4.7

Total number of A - and A^T -calls N_A , CPU time (in seconds) and relative error with dynamic range 60 dB

method	$\epsilon : 10^{-2}$			$\epsilon : 10^{-4}$			$\epsilon : 10^{-6}$		
	time	N_A	rerr	time	N_A	rerr	time	N_A	rerr
ADMM	7.44	599	1.901e-03	13.5	980	2.501e-06	18.7	1403	2.913e-08
ASSN	5.48	449	1.317e-03	9.17	740	1.922e-06	10.2	802	1.930e-08
SPGL1	55.3	2367	5.020e-03	70.7	2978	5.017e-03	89.4	3711	5.017e-03

TABLE 4.8

Total number of A - and A^T -calls N_A , CPU time (in seconds) and relative error with dynamic range 80 dB

method	$\epsilon : 10^{-2}$			$\epsilon : 10^{-4}$			$\epsilon : 10^{-6}$		
	time	N_A	rerr	time	N_A	rerr	time	N_A	rerr
ADMM	7.8	592	5.384e-04	13.8	1040	2.481e-06	17.7	1405	2.350e-08
ASSN	4.15	344	5.194e-04	7.92	618	1.205e-06	8.74	702	5.616e-09
SPGL1	32.2	1368	4.862e-04	56.1	2396	4.859e-04	67.4	2840	4.859e-04

numbers of A - and A^T -calls N_A and CPU time to compare the efficiency among different solvers. We also list the relative error so that we can compare ADMM, ASSN with SPGL1. These numerical results are reported in Tables 4.5 - 4.8. The performance of ASSN is close to ADMM for tolerance 10^{-2} and is much better for tolerance 10^{-4} and 10^{-6} independent of dynamic ranges. For all test problems, SPGL1 can only obtain a low accurate solution. It may be improved if the parameters are further tuned.

Figures 4.3 and 4.4 illustrate the residual history with respect to the total number of A - and A^T -calls N_A and the total number of iterations. SPGL1 is omitted since it cannot converge for a high accuracy. The figures show that ASSN has a similar convergent rate as ADMM in the initial stage but it achieves a faster convergent rate later, in particular, for a high accuracy.

5. Conclusion. The purpose of this paper is to study second-order type methods for solving composite convex programs based on fixed-point mappings induced from many operator splitting approaches such as the FBS and DRS methods. The semi-smooth Newton method is theoretically guaranteed to converge to a global solution from an arbitrary initial point and achieve a fast convergent rate by using an adapt strategy on switching the projection steps and Newton steps. Our proposed algorithms are suitable to constrained convex programs when a fixed-point mapping is well-defined. It may be able to bridge the gap between first-order and second-order type methods. They are indeed promising from our preliminary numerical experiments on a number of applications. In particular, quadratic or superlinear convergence is attainable in some examples of Lasso regression and basis pursuit.

There are a number of future directions worth pursuing from this point on, including theoretical analysis and a comprehensive implementation of these second-order algorithms. To improve the performance in practice, the second-order methods can be activated until the first-order type methods reach a good neighborhood of the global optimal solution. Since solving the corresponding system of linear equations is computationally dominant, it is important to explore the structure of the linear system and design certain suitable preconditioners.

Acknowledgements. The authors would like to thank Professor Defeng Sun for the valuable discussions on semi-smooth Newton methods, and Professor Michael Ulbrich and Dr. Andre Milzarek for sharing their code SNF.

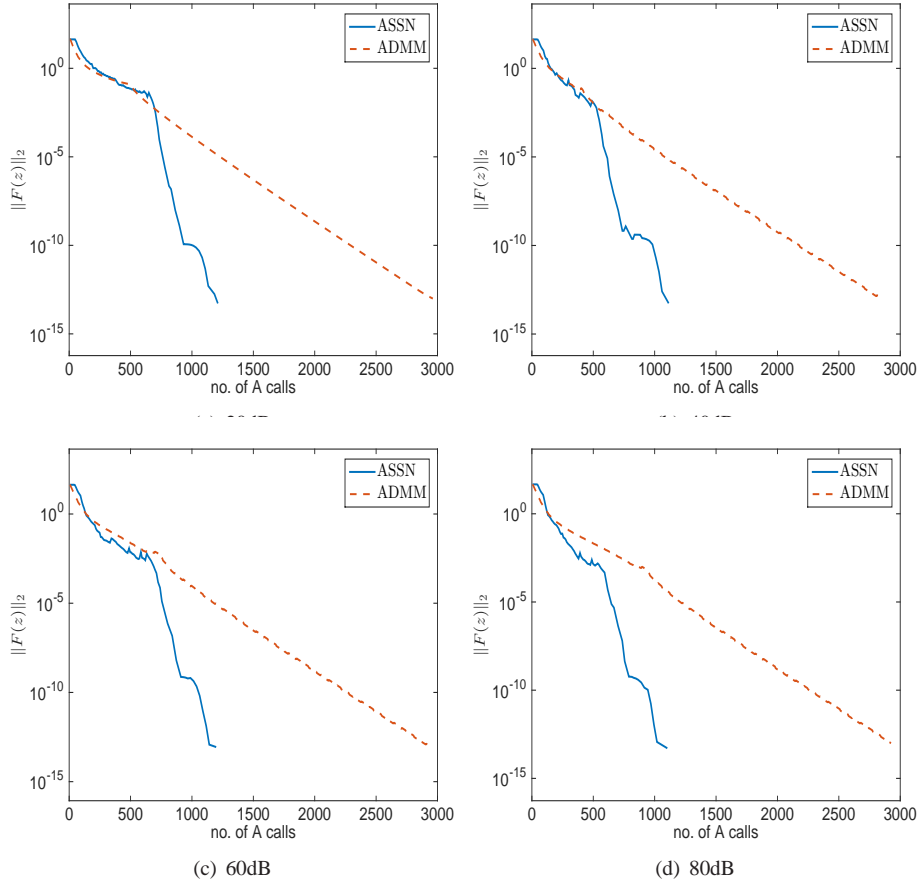


FIG. 4.3. residual history with respect to the total numbers of A - and A^T - calls N_A

REFERENCES

- [1] M. AHOOKHOSH, K. AMINI, AND S. BAHRAMI, *Two derivative-free projection approaches for systems of large-scale nonlinear monotone equations*, Numer. Algorithms, 64 (2013), pp. 21–42.
- [2] H. H. BAUSCHKE AND P. L. COMBETTES, *Convex analysis and monotone operator theory in Hilbert spaces*, Springer, New York, 2011.
- [3] S. BOYD, N. PARIKH, E. CHU, B. PELEATO, AND J. ECKSTEIN, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends in Machine Learning, 3 (2011), pp. 1–122.
- [4] R. H. BYRD, G. M. CHIN, J. NOCEDAL, AND F. OZTOPRAK, *A family of second-order methods for convex ℓ_1 -regularized optimization*. Math. Program., online first, DOI:10.1007/s10107-015-0965-3, 2015.
- [5] C. CHEN, Y. J. LIU, D. SUN, AND K. C. TOH, *A semismooth Newton-CG based dual PPA for matrix spectral norm approximation problems*, Math. Program., 155 (2016), pp. 435–470.
- [6] P. L. COMBETTES AND J.-C. PESQUET, *Proximal splitting methods in signal processing*, in Fixed-point algorithms for inverse problems in science and engineering, vol. 49 of Springer Optim. Appl., Springer, New York, 2011, pp. 185–212.
- [7] D. DAVIS AND W. YIN, *Convergence rate analysis of several splitting schemes*. <http://arxiv.org/abs/1406.4834v3>, 05 2015.
- [8] J. DOUGLAS AND H. H. RACHFORD, *On the numerical solution of heat conduction problems in two and three space variables*, Trans. Amer. Math. Soc., 82 (1956), pp. 421–439.
- [9] D. DRUSVYATSKIY AND A. S. LEWIS, *Error bounds, quadratic growth, and linear convergence of proximal methods*. <http://arxiv.org/abs/1602.06661>, 02 2016.
- [10] J. ECKSTEIN AND D. P. BERTSEKAS, *On the Douglas-Rachford splitting method and the proximal point*

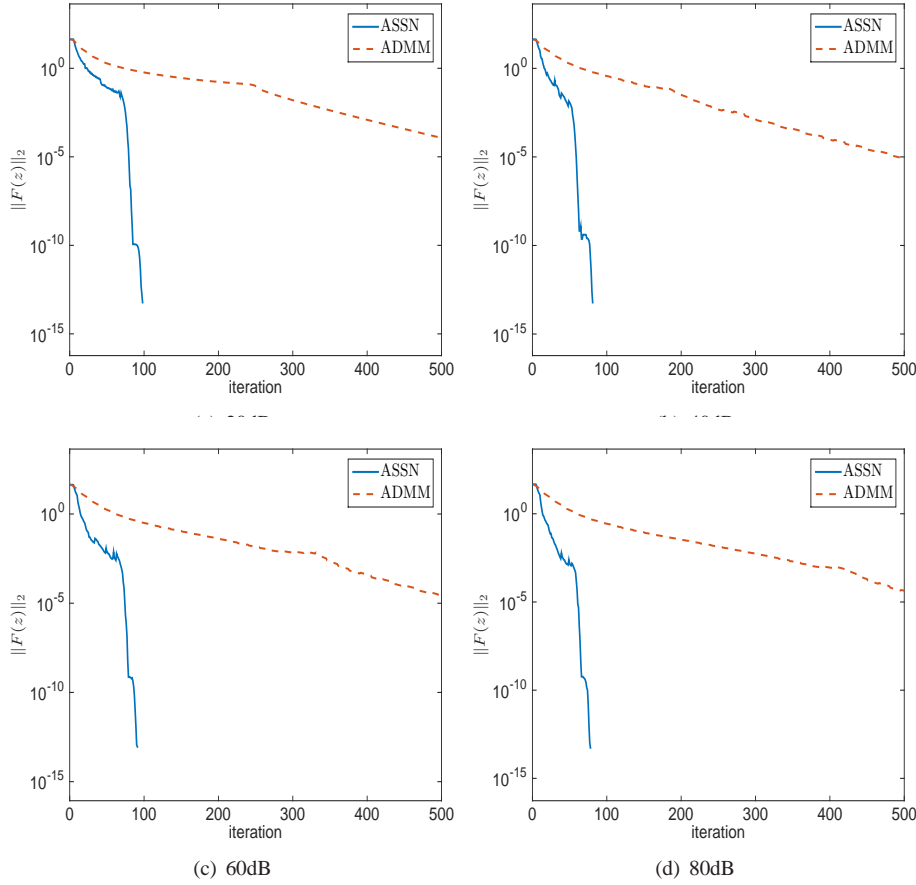


FIG. 4.4. residual history with respect to the total numbers of iteration

- algorithm for maximal monotone operators, *Math. Program.*, 55 (1992), pp. 293–318.
- [11] F. FACCINEI AND J.-S. PANG, *Finite-dimensional variational inequalities and complementarity problems. Vol. I*, Springer-Verlag, New York, 2003.
- [12] ———, *Finite-dimensional variational inequalities and complementarity problems. Vol. II*, Springer-Verlag, New York, 2003.
- [13] J. Y. FAN AND Y. X. YUAN, *On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption*, *Computing*, 74 (2005), pp. 23–39.
- [14] D. GABAY AND B. MERCIER, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, *Computers and Mathematics with Applications*, 2 (1976), pp. 17–40.
- [15] R. GLOWINSKI AND A. MARROCCO, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes de Dirichlet non linéaires*, *Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge Anal. Numér.*, 9 (1975), pp. 41–76.
- [16] R. GRIESSE AND D. A. LORENZ, *A semismooth Newton method for Tikhonov functionals with sparsity constraints*, *Inverse Problems*, 24 (2008), pp. 035007, 19.
- [17] 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.
- [18] D. HAN, D. SUN, AND L. ZHANG, *Linear rate convergence of the alternating direction method of multipliers for convex composite quadratic and semi-definite programming*. <http://arxiv.org/abs/1508.02134>, 8 2015.
- [19] J. D. LEE, Y. SUN, AND M. A. SAUNDERS, *Proximal Newton-type methods for minimizing composite functions*, *SIAM J. Optim.*, 24 (2014), pp. 1420–1443.
- [20] Q. LI AND D. H. LI, *A class of derivative-free methods for large-scale nonlinear monotone equations*, *IMA J. Numer. Anal.*, 31 (2011), pp. 1625–1635.
- [21] P.-L. LIONS AND B. MERCIER, *Splitting algorithms for the sum of two nonlinear operators*, *SIAM J. Numer.*

- Anal., 16 (1979), pp. 964–979.
- [22] Z. Q. LUO AND P. TSENG, *Error bounds and convergence analysis of feasible descent methods: a general approach*, Ann. Oper. Res., 46 (1993), pp. 157–178.
 - [23] R. MIFFLIN, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 15 (1977), pp. 959–972.
 - [24] A. MILZAREK AND M. ULBRICH, *A semismooth Newton method with multidimensional filter globalization for l_1 -optimization*, SIAM J. Optim., 24 (2014), pp. 298–333.
 - [25] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, second ed., 2006.
 - [26] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative solution of nonlinear equations in several variables*, Academic Press, New York-London, 1970.
 - [27] J.-S. PANG AND L. Q. QI, *Nonsmooth equations: motivation and algorithms*, SIAM J. Optim., 3 (1993), pp. 443–465.
 - [28] P. PATRINOS, L. STELLA, AND A. BEMPORAD, *Forward-backward truncated Newton methods for convex composite optimization*. <http://arxiv.org/abs/1402.6655>, 02 2014.
 - [29] L. Q. QI, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.
 - [30] L. Q. QI AND J. SUN, *A nonsmooth version of Newton’s method*, Math. Programming, 58 (1993), pp. 353–367.
 - [31] R. T. ROCKAFELLAR AND R. J.-B. WETS, *Variational analysis*, Springer-Verlag, Berlin, 1998.
 - [32] S. SCHOLTES, *Introduction to piecewise differentiable equations*, Springer Briefs in Optimization, Springer, New York, 2012.
 - [33] A. SHAPIRO, *Directionally nondifferentiable metric projection*, J. Optim. Theory Appl., 81 (1994), pp. 203–204.
 - [34] M. V. SOLODOV AND B. F. SVAITER, *A globally convergent inexact Newton method for systems of monotone equations*, in Reformulation: nonsmooth, piecewise smooth, semismooth and smoothing methods (Lausanne, 1997), M. Fukushima and L. Qi, eds., vol. 22, Kluwer Academic Publishers, Dordrecht, 1999, pp. 355–369.
 - [35] D. SUN AND J. SUN, *Semismooth matrix-valued functions*, Math. Oper. Res., 27 (2002), pp. 150–169.
 - [36] P. TSENG, *Approximation accuracy, gradient methods, and error bound for structured convex optimization*, Math. Program., 125 (2010), pp. 263–295.
 - [37] M. ULBRICH, *Semismooth Newton methods for variational inequalities and constrained optimization problems in function spaces*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, 2011.
 - [38] E. VAN DEN BERG AND M. P. FRIEDLANDER, *Probing the Pareto frontier for basis pursuit solutions*, SIAM J. Sci. Comput., 31 (2008), pp. 890–912.
 - [39] Z. WEN, W. YIN, D. GOLDFARB, AND Y. ZHANG, *A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation*, SIAM J. Sci. Comput., 32 (2010), pp. 1832–1857.
 - [40] 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.
 - [41] N. YAMASHITA AND M. FUKUSHIMA, *On the rate of convergence of the Levenberg-Marquardt method*, in Topics in numerical analysis, vol. 15 of Comput. Suppl., Springer, Vienna, 2001, pp. 239–249.
 - [42] X. Y. ZHAO, D. SUN, AND K. C. TOH, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. Optim., 20 (2010), pp. 1737–1765.
 - [43] Y.-B. ZHAO AND D. LI, *Monotonicity of fixed point and normal mappings associated with variational inequality and its application*, SIAM J. Optim., 11 (2001), pp. 962–973.
 - [44] G. ZHOU AND K. C. TOH, *Superlinear convergence of a Newton-type algorithm for monotone equations*, J. Optim. Theory Appl., 125 (2005), pp. 205–221.
 - [45] W. J. ZHOU AND D. H. LI, *A globally convergent BFGS method for nonlinear monotone equations without any merit functions*, Math. Comp., 77 (2008), pp. 2231–2240.