

A Regularized Semi-Smooth Newton Method with Projection Steps for Composite Convex Programs

Xiantao Xiao¹ · Yongfeng Li² · Zaiwen Wen³  ·
Liwei Zhang¹

Received: 2 February 2017 / Revised: 14 August 2017 / Accepted: 29 November 2017 /
Published online: 8 December 2017
© Springer Science+Business Media, LLC, part of Springer Nature 2017

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: (1) Many well-known operator splitting methods, such as forward–backward splitting and Douglas–Rachford splitting, actually define a fixed-point mapping; (2) 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. These nonlinear equations may be non-differentiable, but 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.

X. Xiao: Research supported by the Fundamental Research Funds for the Central Universities under the Grant DUT16LK30. Z. Wen: Research supported in part by NSFC Grant 91730302, 11421101, and by the National Basic Research Project under the Grant 2015CB856002. L. Zhang: Research partially supported by NSFC Grants 11571059 and 91330206.

✉ Zaiwen Wen
wenzw@pku.edu.cn

Xiantao Xiao
txiao@dlut.edu.cn

Yongfeng Li
YongfengLi@pku.edu.cn

Liwei Zhang
lwzhang@dlut.edu.cn

¹ School of Mathematical Sciences, Dalian University of Technology, Dalian, China

² School of Mathematical Sciences, Peking University, Beijing, China

³ Beijing International Center for Mathematical Research, Peking University, Beijing, China

Keywords Composite convex programs · Operator splitting methods · Proximal mapping · Semi-smoothness · Newton method

Mathematics Subject Classification 90C30 · 65K05

1 Introduction

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

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

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 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 since they are easy to be implemented in many interesting 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 [18, 26] 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 [4, 16, 22] solve the nonsmooth formulation of the optimality conditions corresponding to the FBS method. In [43], 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. These nonlinear equations are often non-differentiable, but they are semi-smooth due to the properties of the proximal mappings. Hence, we propose a regularized semi-smooth Newton method to solve the system of nonlinear equations. Although the semi-smooth Newton method had been extensively studied in 1990s [9, 25, 27, 28, 34, 37], our usage is still different in several aspects. Firstly, our regularization parameter is updated by a self-adaptive strategy similar to the trust region algorithms. The regularization term is important since the generalized Jacobian matrix corresponding to monotone equations may only be positive semidefinite. Secondly, our approach keeps to use the semi-smooth Newton step as much as possible if the residual is reduced sufficiently comparing to the last Newton step. Otherwise, a hyperplane projection technique proposed by [32, 33] is called to ensure global convergence to an optimal solution of problem (1.1). 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. Of course, the projection step may be replaced by other globalization techniques for certain problems. Thirdly, the computational cost can be further reduced if the Jacobian matrix is approximated by the limited memory BFGS (L-BFGS) method for some cases.

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 [4, 16, 18, 22, 26] are designed for unconstrained problems. Unlike the methods in [5, 43] 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. Since the hyperplane projection technique in [32, 33] often slows down the numerical convergence of the Newton steps, we only activate this step when it is necessary and the numerical efficiency is still determined by the Newton steps. 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 Sect. 2, we review a few popular operator splitting methods, derive their equivalent fixed-point iterations and discuss their semi-smooth and monotone properties. We propose a semi-smooth Newton method and establish its convergence results in Sect. 3. Numerical results on a number of applications are presented in Sect. 4. Finally, we conclude this paper in Sect. 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

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

The Fenchel conjugate function f^* of f is

$$f^*(y) := \sup_{x \in \mathbb{R}^n} \{x^T y - f(x)\}. \quad (1.3)$$

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

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

The mapping F is called *nonexpansive (contractive)* if F is Lipschitz continuous with constant $L \leq 1$ ($L < 1$), and *firmly nonexpansive* if

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

The mapping F 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

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

The operator T is called α -*averaged* ($\alpha \in (0, 1]$) if there exists a nonexpansive operator R such that $T = (1 - \alpha)I + \alpha R$.

2 Preliminaries

In this section, we review the fixed-point characterizations of some operator splitting algorithms and discuss some important properties of the associated fixed-point mappings. This section consists of three subsections. Section 2.1 reviews some operator splitting algorithms for problem (1.1), including FBS, DRS, and ADMM. These algorithms are well studied in the literature, see [2, 6, 7, 12] for example. Most of the operator splitting algorithms can also be interpreted as fixed-point algorithms derived from certain optimality conditions. Then, we present that the fixed-point mappings derived from the operator splitting algorithms are semi-smooth and monotone in Sects. 2.2 and 2.3, respectively.

2.1 Operator Splitting and Fixed-Point Algorithms

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

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

where $t > 0$ is the step size. Define the following operator

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

Then FBS can be viewed as a fixed-point iteration

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

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

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

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

$$z^{k+1} = z^k + y^{k+1} - x^{k+1}. \tag{2.6}$$

The algorithm is traced back to [10, 11, 20] to solve partial differential equations (PDEs). The fixed-point iteration characterization of DRS is in the form of

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

where

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

Consider a linear constrained program:

$$\begin{aligned} & \min_{x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{R}^{n_2}} f_1(x_1) + f_2(x_2) \\ & \text{subject to } A_1x_1 + A_2x_2 = b, \end{aligned} \tag{2.9}$$

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

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

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 and f_2 are convex. It is widely known that the DRS iteration for dual problem (2.10) is the ADMM [14, 15]. 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).

2.2 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 2.1 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 [21] for real-valued functions and extended to vector-valued mappings by Qi and Sun [28].

Definition 2.2 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 $J \in \partial F(x + d)$,

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

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 $J \in \partial F(x + d)$,

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

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. Examples of semi-smooth functions are thoroughly studied in [12,37].

The basic properties of proximal mapping is well documented in textbooks such as [2,29]. 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 [31].

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

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

For a comprehensive study on piecewise \mathcal{C}^k functions, the readers are referred to [30]. 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 [26, 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 [29, Example 12.31] and hence strongly semi-smooth. The projections over symmetric cones are proved to be strongly semi-smooth in [35].

2.3 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 and nonexpansive property of F play critical roles in our proposed method.

- Proposition 2.3** (i) *Suppose that ∇h is cocoercive with $\beta > 0$, then $F_{FBS} = I - T_{FBS}$ is monotone and $\frac{2\beta}{4\beta-\gamma}$ -averaged 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 $h \in C^2$, $H(x) := \nabla^2 h(x)$ 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 $F_{DRS} := I - T_{DRS}$ is monotone and firmly nonexpansive hence $\frac{1}{2}$ -averaged.
- (v) If T is α -averaged ($\alpha \in (0, 1]$), then $F = I - T$ is $\frac{1}{2\alpha}$ -cocoercive.

Proof Items (i) and (ii) are well known in the literature, see [8,44] for example.

(iii) From the mean value theorem, for any $x, y \in \mathbb{R}^n$, one has that

$$\nabla h(x) - \nabla h(y) = \bar{H}(x, y)(x - y),$$

where $\bar{H}(x, y) = \int_0^1 H(y + s(x - y))ds$ and thus $\lambda_{\max}(\bar{H}(x, y)) \leq \bar{\lambda}$. Hence, we obtain

$$\|\nabla h(x) - \nabla h(y)\|_2^2 = \langle \bar{H}(x, y)(x - y), \bar{H}(x, y)(x - y) \rangle \leq \bar{\lambda} \langle x - y, \nabla h(x) - \nabla h(y) \rangle,$$

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 [20]. Therefore, F_{DRS} is firmly nonexpansive and hence monotone [2, Proposition 4.2].

(v) The proof simply follows from the definition of the α -average of T . □

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 [44] 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. Finally, we introduce an useful lemma on the positive semidefinite property of the subdifferential of the monotone mapping, see [17, Proposition 2.3].

Lemma 2.4 *For a Lipschitz continuous mapping $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, F is monotone if and only if each element of $\partial_B F(x)$ is positive semidefinite for any $x \in \mathbb{R}^n$.*

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

$$F(z) = 0, \tag{3.1}$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is 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. Let $Z^* := \{z \in \mathbb{R}^n : F(z) = 0\}$ denote the solution set to nonlinear equations (3.1). It is known that Z^* is a closed convex set if the mapping F is maximal monotone.

3.1 A Regularized Semi-Smooth Newton Method with Projection Steps

The system of monotone equations has various applications [1, 19, 24, 32, 46]. Inspired by a pioneer work [32], a class of iterative methods for solving nonlinear (smooth) monotone equations were proposed in recent years [1, 19, 46]. In [32], 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 [45] to solve monotone equations without nonsingularity assumption.

The main concept in [32] is introduced as follows starting from an initial point z^0 . For an iterate z^k , let d^k be a 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

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

strictly separates z^k from the solution set Z^* . Based on this fact, it was developed in [32] 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 2.4 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

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

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

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

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

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

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

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

If $\|F(u^k)\|_2$ is sufficiently decreased, we take a Newton step. Specifically, let $\xi_0 = \|F(z^0)\|_2$. When $\|F(u^k)\|_2 \leq \nu \xi_k$ with $0 < \nu < 1$, we call the Newton step is *successful* and set

$$z^{k+1} = u^k, \xi_{k+1} = \|F(u^k)\|_2 \text{ and } \lambda_{k+1} = \lambda_k. \text{ [Newton step]} \tag{3.5}$$

Otherwise, we take a safeguard step as follows. Define a ratio

$$\rho_k = \frac{-\langle F(u^k), d^k \rangle}{\|d^k\|_2^2}. \tag{3.6}$$

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, we take a hyperplane projection step when the residual of the projection step is non-increasing and take a fixed-point iteration when it is increasing. In summary, we set

$$z^{k+1} = \begin{cases} v^k, & \text{if } \rho_k \geq \eta_1 \text{ and } \|F(v^k)\|_2 \leq \|F(z^k)\|_2, \text{ [projection step]} \\ w^k, & \text{if } \rho_k \geq \eta_1 \text{ and } \|F(v^k)\|_2 > \|F(z^k)\|_2, \text{ [fixed-point step]} \\ z^k, & \text{if } \rho_k < \eta_1, \text{ [unsuccessful step]} \end{cases} \tag{3.7}$$

where

$$v^k = z^k - \frac{\langle F(u^k), z^k - u^k \rangle}{\|F(u^k)\|_2^2} F(u^k), \quad w^k = z^k - \beta F(z^k), \beta \in \left(0, \frac{1}{\alpha}\right). \tag{3.8}$$

The parameters ξ_{k+1} and λ_{k+1} are updated as

$$\xi_{k+1} = \xi_k, \quad \lambda_{k+1} \in \begin{cases} (\underline{\lambda}, \lambda_k), & \text{if } \rho_k \geq \eta_2, \\ [\lambda_k, \gamma_1 \lambda_k], & \text{if } \eta_1 \leq \rho_k < \eta_2, \\ (\gamma_1 \lambda_k, \gamma_2 \lambda_k], & \text{otherwise,} \end{cases} \tag{3.9}$$

where $\underline{\lambda} > 0$ is a small positive constant. These parameters determine how aggressively the regularization parameter is increased when an iteration is unsuccessful or it is decreased when $\rho_k \geq \eta_2$. The complete approach to solve (3.1) is summarized in Algorithm 1.

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

- 1 Give $0 < \tau, v < 1, 0 < \eta_1 \leq \eta_2 < 1$ and $1 < \gamma_1 \leq \gamma_2$;
 - 2 Choose z^0 and $\varepsilon > 0$. Set $k = 0$ and $\xi_0 = \|F(z^0)\|_2$;
 - 3 **while not “converged” do**
 - 4 Select $J_k \in \partial_B F(z^k)$;
 - 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 ρ_k as in (3.6);
 - 7 If $\|F(u^k)\|_2 \leq v\xi_k$, update z^{k+1}, ξ_{k+1} and λ_{k+1} according to (3.5). Otherwise, set them according to (3.7) and (3.9), respectively;
 - 8 Set $k = k + 1$;
-

3.2 Global Convergence

We make the following assumption.

Assumption 3.1 The set Z^* of optimal solutions is nonempty. The function $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is monotone and the corresponding fixed point operator $T = I - F$ is α -averaged with $\alpha \in (0, 1]$.

The assumption that T is α -averaged implies that F is Lipschitz continuous with modulus $L \leq 2\alpha$. Hence, we have $\|J_k\| \leq L$ for any $k \geq 0$ and any $J_k \in \partial_B F(z^k)$. Since the Lipschitz continuity of F is sufficient to obtain global convergence, the semi-smoothness of F actually is not required in this subsection.

The following lemma demonstrates that the distance from z^k to Z^* decreases in a projection step. The proof follows directly from [32, Lemma 2.1], and it is omitted.

Lemma 3.2 For any $z^* \in Z^*$ and any projection step, indexed by say k , we have that

$$\|z^{k+1} - z^*\|_2^2 \leq \|z^k - z^*\|_2^2 - \|z^{k+1} - z^k\|_2^2. \tag{3.10}$$

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

$$\begin{aligned} \mathcal{K}_N &:= \{k \geq 0 \mid \|F(u^k)\|_2 \leq \nu\xi_k\}, \\ \mathcal{K}_P &:= \{k \geq 0 \mid \rho_k \geq \eta_1, \|F(u^k)\|_2 > \nu\xi_k, \|F(v^k)\|_2 \leq \|F(z^k)\|_2\}, \\ \mathcal{K}_F &:= \{k \geq 0 \mid \rho_k \geq \eta_1, \|F(u^k)\|_2 > \nu\xi_k, \|F(v^k)\|_2 > \|F(z^k)\|_2\}, \end{aligned}$$

and

$$\mathcal{K}_S := \{k \geq 0 \mid \|F(u^k)\|_2 \leq \nu\xi_k \text{ or } \rho_k \geq \eta_1\}.$$

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

Lemma 3.3 Suppose that Assumption 3.1 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 . By the definition of “successful” and “unsuccessful”, we have that $\rho_k < \eta_1$ for all $k > k_0$. Therefore, from (3.7) and (3.9), one has that $z^{k_0+i} = z^{k_0+1} := z^*$, for all $i \geq 1$ and additionally $\lambda_k \rightarrow \infty$. We proceed by contradiction. Suppose that $\phi := \|F(z^*)\|_2 > 0$. 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

$$\lambda_k \geq \frac{\eta_2 + L}{(1 - \tau)\phi}.$$

Then, it yields that

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

The second line above is from that J_k is positive semidefinite and the residual condition (3.4). The last inequality is obtained by $\|F^k\|_2 = \|F(z^*)\|_2 = \phi$ for $k > k_0$. Further, we obtain

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

where the first inequality is derived by (3.11) and the second inequality comes from L -Lipschitz continuity of F . Hence, we have $\rho_k \geq \eta_2$, which generates a successful iteration and yields a contradiction. This completes the proof. \square

We are now ready to prove the main global convergence result.

Theorem 3.4 *If Assumption 3.1 holds, then we have $\lim_{k \rightarrow \infty} \|F(z^k)\|_2 = 0$.*

Proof If the index set \mathcal{K}_S is finite, the convergence is directly from Lemma 3.3. Thus, we only need to consider the situation that the index set \mathcal{K}_S is infinite. The main task is to prove that the full sequence $\{\|F(z^k)\|_2\}_{k \geq 0}$ converges to zero in the following two cases.

Case (i): \mathcal{K}_N is finite.

Without loss of generality, we ignore \mathcal{K}_N and assume that $\mathcal{K}_S = \mathcal{K}_P \cup \mathcal{K}_F$ in this case. Let z^* be any point in solution set Z^* . For any $k \in \mathcal{K}_F$, we have that $z^{k+1} = z^k - \beta F(z^k)$ and

$$\begin{aligned} \|z^k - z^*\|_2^2 - \|z^{k+1} - z^*\|_2^2 &= 2\beta \langle F(z^k), z^k - z^* \rangle - \beta^2 \|F(z^k)\|_2^2 \\ &\geq \beta \left(\frac{1}{\alpha} - \beta \right) \|F(z^k)\|_2^2 \geq 0, \end{aligned} \tag{3.12}$$

where the first inequality is due to F is $\frac{1}{2\alpha}$ -cocoercive and the item (v) in Proposition 2.3, and the last inequality is because of the choice of β . By Lemma 3.2, for any $k \in \mathcal{K}_P$, it yields that

$$\|z^k - z^*\|_2^2 - \|z^{k+1} - z^*\|_2^2 \geq \|z^{k+1} - z^k\|_2^2. \tag{3.13}$$

Therefore, the whole sequence $\{\|z^k - z^*\|_2\}_{k \geq 0}$ is non-increasing and convergent, and the sequence $\{z^k\}_{k \geq 0}$ is bounded. 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 $\{d^k\}_{k \geq 0}$ is bounded, and $\{u^k\}_{k \geq 0}$ is also bounded. By using the continuity of F , there exists a constant $c_1 > 0$ such that

$$\|F(u^k)\|_2^{-1} \geq c_1, \quad \text{for any } k \geq 0.$$

Using (3.7) and (3.8), for any $k \in \mathcal{K}_P$, we obtain that

$$\|z^{k+1} - z^k\|_2 = \frac{-\langle F(u^k), d^k \rangle}{\|F(u^k)\|_2} \geq c_1 \rho_k \|d^k\|_2^2 \geq c_1 \eta_1 \|d^k\|_2^2. \tag{3.14}$$

We next consider either

$$\liminf_{k \rightarrow \infty} \|F^k\|_2 = 0 \quad \text{or} \quad \liminf_{k \rightarrow \infty} \|F^k\|_2 = c_2 > 0.$$

If $\liminf_{k \rightarrow \infty} \|F^k\|_2 = 0$, the continuity of F and the boundedness of $\{z^k\}_{k \geq 0}$ imply that the sequence $\{z^k\}_{k \geq 0}$ 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) and (3.13). Then $\{z^k\}_{k \geq 0}$ converges to \hat{z} and $\{\|F(z^k)\|_2\}_{k \geq 0}$ converges to zero.

If $\liminf_{k \rightarrow \infty} \|F^k\|_2 = c_2 > 0$, by using the continuity of F and the boundedness of $\{z^k\}_{k \geq 0}$ again, there exist constants $c_3 > c_4 > 0$ such that

$$c_4 \leq \|F^k\|_2 \leq c_3, \quad \text{for all } k \geq 0.$$

We now show that $\{\lambda_k\}_{k \geq 0}$ is bounded above. If λ_k is large enough such that $\|d^k\|_2 \leq 1$ and

$$\lambda_k \geq \frac{\eta_2 + L}{(1 - \tau)c_4},$$

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

$$\|F^k\|_2 \leq \|(J_k + \lambda_k \|F^k\|_2 I)d^k\|_2 + \|r^k\|_2 \leq (L + (1 + \tau)c_3\bar{\lambda})\|d^k\|_2. \tag{3.15}$$

Hence, it follows from (3.13), (3.14) and (3.15) that there exists a constant $c_5 > 0$ such that for any $k \in \mathcal{K}_P$,

$$\|z^k - z^*\|_2^2 - \|z^{k+1} - z^*\|_2^2 \geq c_5 \|F(z^k)\|_2^4. \tag{3.16}$$

Then, by (3.12) and (3.16), we conclude that $\{\|F(z^k)\|_2\}_{k \geq 0}$ converges to zero, which yields a contradiction to the assumption $c_2 > 0$. The convergence of $\{z^k\}_{k \geq 0}$ and $\{\|F(z^k)\|_2\}_{k \geq 0}$ is established in the case (i).

Case (ii): K_N is infinite.

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 $0 < \nu < 1$ for any $i \geq 1$, we have that the sequence $\{\|F(z^{k_i})\|_2\}$ converges to zero. For any $k \in \mathcal{K}_P$, we have $\|F(z^{k+1})\|_2 \leq \|F(z^k)\|_2$ from the updating rule (3.7). For any $k \in \mathcal{K}_F$, we have that $z^{k+1} = (I - \beta F)(z^k) = [(1 - \alpha\beta)I + \alpha\beta R](z^k)$ with $\alpha\beta < 1$. Since R is nonexpansive, from [7, Theorem 1], we have that $\|(R - I)(z^{k+1})\|_2 \leq \|(R - I)(z^k)\|_2$. By noticing that $F(z^k) = \alpha(I - R)(z^k)$, it follows that $\|F(z^{k+1})\|_2 \leq \|F(z^k)\|_2$, for any $k \in \mathcal{K}_F$. Moreover, for any $k \notin \mathcal{K}_N$, there exists an index i such that $k_i < k + 1 < k_{i+1}$, and hence $\|F(z^{k+1})\|_2 \leq \|F(z^{k_i})\|_2$. Therefore, the whole sequence $\{\|F(z^k)\|_2\}_{k \leq 0}$ converges to 0, which completes the proof. \square

Some comments to Theorem 3.4 are given as follows.

- (i) In our arguments, we do not assume that $\{z^k\}$ is bounded. In fact, the subsequence $\{z^k\}_{k \in \mathcal{K}_P \cup \mathcal{K}_F}$ is bounded because of (3.12) and (3.13). Hence, the existence of accumulation points is up to the boundedness of the subsequence $\{z^k\}_{k \in \mathcal{K}_N}$. Then any accumulation point of $\{z^k\}$ converges to some point \bar{z} such that $F(\bar{z}) = 0$.
- (ii) If the global error bound condition is satisfied at \bar{z} , i.e., there exists a constant $c > 0$ such that

$$c\|z - \bar{z}\| \leq \|F(z)\|_2, \quad \forall z \in \mathbb{R}^n, \tag{3.17}$$

the whole sequence $\{z^k\}$ converges to \bar{z} . Consider the case that the mapping $F(x)$ is derived from the fixed point mapping with respect to the FBS, where the function h in (2.2) is strongly convex. Then the optimal solution z^* is unique and the global error bound condition is satisfied at z^* [36, Theorem 4].

As is already shown, the global convergence of our Algorithm is essentially guaranteed by the projection step and the fixed-point step. However, by noticing that (3.8) 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 and fixed-point step are always performed. Hence, our modification (3.7) is practically meaningful. Moreover, we will next prove that the projection step and fixed-point step will never be performed when the iterate is close enough to a solution under some generalized nonsingular conditions.

3.3 Fast Local Convergence

By the global convergence analysis, we can assume that z^* is an accumulation point of the sequence $\{z^k\}$ generated by Algorithm 1 such that $F(z^*) = 0$. We also make the following assumption.

Assumption 3.5 For the mapping F defined in (3.1), the mapping F is semi-smooth and BD-regular at z^* , which means that all elements in $\partial_B F(z^*)$ are nonsingular.

The semi-smoothness of the mapping F often holds since the proximal mapping is (strongly) semi-smooth in many interesting applications. The BD-regularity is a common assumption in the analysis of the local convergence of nonsmooth methods. When F is BD-regular at z^* , from [27, Proposition 2.5] and [25, Proposition 3], there exists some constants $c_6 > 0$, $\kappa > 0$ and a neighborhood $N(z^*, \varepsilon_0)$ such that for any $y \in N(z^*, \varepsilon_0)$ and $J \in \partial_B F(y)$, it holds that:

- (1) z^* is an isolated solution;
- (2) J is nonsingular and $\|J^{-1}\| \leq c_6$;
- (3) the local error bound condition holds for $F(z)$ on $N(z^*, \varepsilon_0)$, that is $\|y - z^*\|_2 \leq \kappa \|F(y)\|_2$.

Note that, the conditions that z^* is isolated and $\|z^k - z^*\|_2 \leq \kappa \|F^k\|_2$ (for k large enough) indicate that the whole sequence $\{z^k\}$ converges to z^* .

We next show that Algorithm 1 turns into a Newton-type method in a neighborhood of z^* and achieves a locally fast convergence.

Theorem 3.6 *Suppose that Assumption 3.5 holds and the regularization parameter λ_k is bounded above by $\bar{\lambda}$ for any $z^k \in N(z^*, \varepsilon_1)$ with $\varepsilon_1 := \min\{\varepsilon_0, 1/(2Lc_6\tau\bar{\lambda})\}$. Then for sufficiently large k , we have $\|F(u^k)\|_2 \leq \nu \|F(z^k)\|_2$ and $z^{k+1} = u^k$, and $\{z^k\}$ converges superlinearly to z^* . If F is strongly semi-smooth at z^* , then $\{z^k\}$ converges quadratically to z^* .*

Proof It follows from the L -Lipschitz continuity of F that $\|F^k\|_2 \leq L\|z^k - z^*\|_2 \leq L\varepsilon_1$ for any $z^k \in N(z^*, \varepsilon_1)$. Hence the definition of ε_1 implies that

$$c_6\tau\bar{\lambda}\|F^k\|_2 \leq 1/2. \tag{3.18}$$

From (3.2), (3.4), (3.18) and $\|J_k^{-1}\| \leq c_6$, we obtain for a Newton step that

$$\begin{aligned} \|d^k\|_2 &\leq \|(J_k + \mu_k I)^{-1} F^k\|_2 + \|(J_k + \mu_k I)^{-1} r^k\|_2 \\ &\leq c_6 L \|z^k - z^*\|_2 + c_6\tau\bar{\lambda}\|F^k\|_2 \|d^k\|_2 \\ &\leq 2c_6 L \|z^k - z^*\|_2, \end{aligned} \tag{3.19}$$

where the last inequality is derived from a rearrangement of the second inequality. A direct calculation gives

$$\begin{aligned} \|u^k - z^*\|_2 &= \|z^k + d^k - z^*\|_2 \\ &= \|z^k + (J_k + \mu_k I)^{-1}(F^k + (J_k + \mu_k I)d^k - F^k) - z^*\|_2 \\ &\leq \|z^k - z^* - (J_k + \mu_k I)^{-1} F^k\|_2 + \|(J_k + \mu_k I)^{-1} \cdot \|F^k + (J_k + \mu_k I)d^k\|_2 \\ &\leq \|(J_k + \mu_k I)^{-1}\| \cdot [\|F^k - F(z^*) - J_k(z^k - z^*)\|_2 + \mu_k \|z^k - z^*\|_2 + \|r^k\|_2]. \end{aligned}$$

Using $\|(J_k + \mu_k I)^{-1}\| \leq c_6$, $\mu_k = \lambda_k \|F^k\|_2$ and $\|r^k\|_2 \leq \tau\lambda_k \|F^k\|_2 \|d^k\|_2$, we have

$$\begin{aligned} \|u^k - z^*\|_2 &\leq c_6(\|F^k - F(z^*) - J_k(z^k - z^*)\|_2 \\ &\quad + \bar{\lambda}\|F^k\|_2 \|z^k - z^*\|_2 + \tau\bar{\lambda}\|F^k\|_2 \|d^k\|_2). \end{aligned} \tag{3.20}$$

By the L -Lipschitz continuity of F and (3.19), we obtain

$$\bar{\lambda} \|F^k\|_2 \|z^k - z^*\|_2 + \tau \bar{\lambda} \|F^k\|_2 \|d^k\|_2 \leq L \bar{\lambda} (1 + 2c_6 L \tau) \|z^k - z^*\|_2^2. \tag{3.21}$$

The semi-smoothness of F at z^* implies

$$\|F^k - F(z^*) - J_k(z^k - z^*)\|_2 = o(\|z^k - z^*\|_2), \text{ as } \|z^k - z^*\| \rightarrow 0,$$

which, together with (3.20) and (3.21) shows that $\|u^k - z^*\|_2 = o(\|z^k - z^*\|_2)$. Hence, we have $L\|u^k - z^*\|_2 \leq \frac{\nu}{\kappa} \|z^k - z^*\|_2$ for sufficiently large k . The local error bound condition implies that $\|z^k - z^*\|_2 \leq \kappa \|F(z^k)\|_2$ for sufficiently large k . Together with the Lipschitz continuity of F , we obtain for sufficiently large k that

$$\|F(u^k)\|_2 \leq L\|u^k - z^*\|_2 \leq \frac{\nu}{\kappa} \|z^k - z^*\|_2 \leq \nu \|F(z^k)\|_2.$$

Then the updating rule (3.5) yields $\|F(u^k)\|_2 \leq \nu \|F(z^k)\|_2$ and $z^{k+1} = u^k$.

When F is strongly semi-smooth, the quadratic convergence $\|u^k - z^*\|_2 = O(\|z^k - z^*\|_2^2)$ is established because of the property $\|F^k - F(z^*) - J_k(z^k - z^*)\|_2 = O(\|z^k - z^*\|_2^2)$. The proof is completed. \square

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 [22, 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.

3.4 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 [46] 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 [46]. 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

$$(H_k + \mu_k I)d^k = -F^k, \tag{3.22}$$

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 [23] as

$$H_k = H_k^0 - [H_k^0 \mathcal{D}_k \mathcal{F}_k] \begin{bmatrix} \mathcal{D}_k^T H_k^0 \mathcal{D}_k & L_k \\ L_k^T & -S_k \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{D}_k^T (H_k^0)^T \\ \mathcal{F}_k^T \end{bmatrix}, \tag{3.23}$$

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 $\bar{H}_k = H_k^0 + \mu_k I$, $C_k = [H_k^0 \mathcal{D}_k \mathcal{F}_k]$, 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 (3.23).

We should point out that the usage of L-BFGS may not be suitable because the Jacobian matrix may not be symmetric. It seems that no quasi-Newton method can provide a non-symmetric but positive semidefinite approximation. Hence, we only take care of the positive semidefiniteness rather than nonsymmetric property. Fortunately, our numerical experiments show that it works in the ℓ_1 -regularized optimization problem. Of course, one may also try nonsymmetric type of quasi-Newton method.

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

$$\min \mu \|x\|_1 + h(x), \tag{4.1}$$

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

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

where $M(x) \in \partial \mathbf{prox}_{t f}(x - t \nabla h(x))$ 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

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

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 [22], 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

$$J(x) = \begin{pmatrix} t(\partial^2 h(x))_{\mathcal{I}(x)\mathcal{I}(x)} & t(\partial^2 h(x))_{\mathcal{I}(x)\mathcal{O}(x)} \\ 0 & I \end{pmatrix}. \tag{4.3}$$

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

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

The second equation in the above linear system is then solved by the conjugate gradient (CG) method.

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 [22]. ASSN is the proposed semi-smooth Newton method with projection steps (Algorithm 1) and SSNP is the method which only uses the projection steps. ALSB(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 CG method. The parameters of FPC-AS, SpaRSA and SNF are the same as [22].

Table 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 |

The test problems are from [22], 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 $k = \lfloor n/40 \rfloor = 5553$. 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 $\|F(x)\| \leq 10^{-14}$.

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 1, 2, 3, and 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 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 to emphasize that L-BFGS with $m = 1$ is equal to the Hestenes–Stiefel and Polak–Ribière conjugate gradient method with exact line search [23]. Compared with ASSN, SSNP has a slower convergent rate, which implies that our adaptive strategy on switching Newton and projection steps is helpful.

Table 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 |

Table 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 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 |

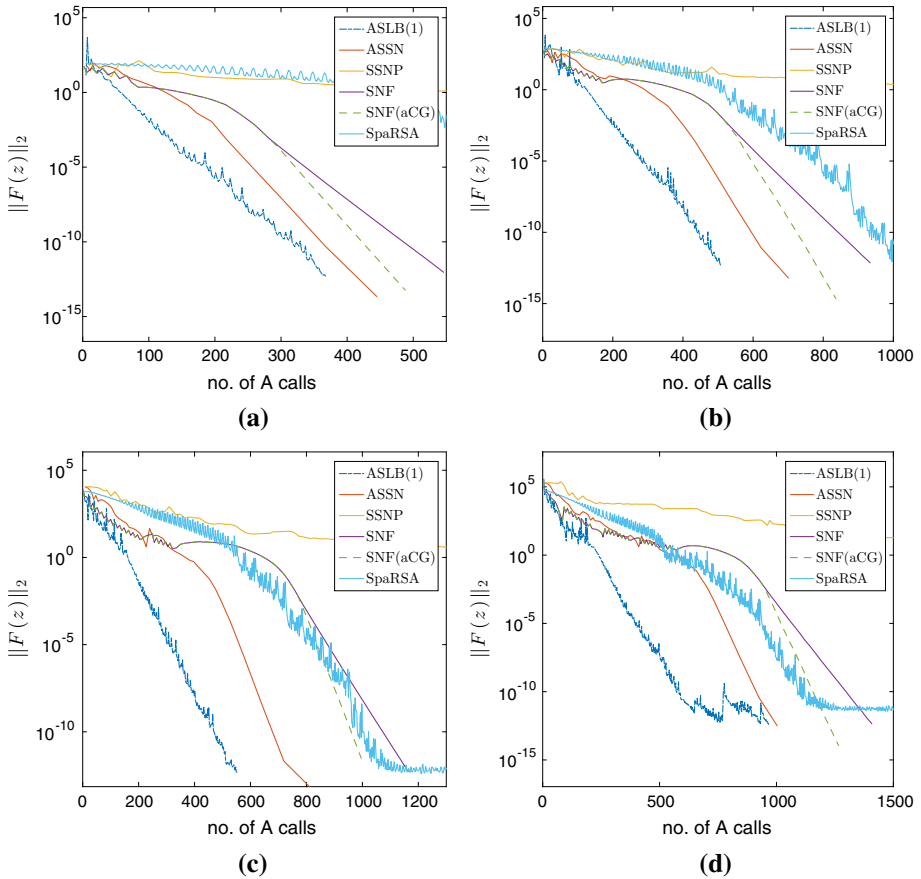


Fig. 1 Residual history with respect to the total numbers of A - and A^T -calls N_A . **a** 20 dB, **b** 40 dB, **c** 60 dB, **d** 80 dB

In particular, ASSN and ASLB have a better performance for high accuracy. Figures 1 and 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 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

$$\min \|x\|_1, \text{ subject to } Ax = b, \tag{4.4}$$

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

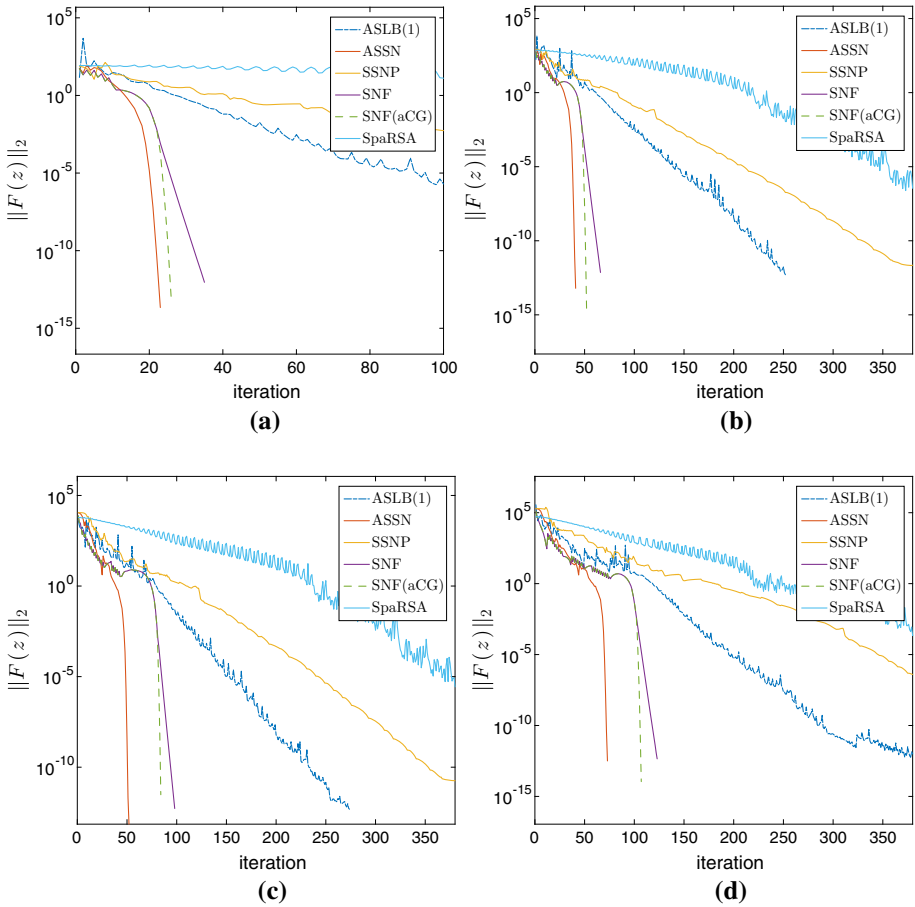


Fig. 2 Residual history with respect to the total numbers of iteration. **a** 20 dB, **b** 40 dB, **c** 60 dB, **d** 80 dB

$$F(z) = \mathbf{prox}_{t_h}(z) - \mathbf{prox}_{t_f}(2\mathbf{prox}_{t_h}(z) - z) = 0. \tag{4.5}$$

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

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

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

$$D = I - A^T A. \tag{4.6}$$

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

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

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

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

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

$$J(z) = M(z) + D(I - 2M(z)). \tag{4.7}$$

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

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

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

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

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, $I - A W H^{-1} A^T = (1 - \frac{1}{1+\mu}) I + A S A^T$. 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

$$A S A^T = \left(\frac{1}{\mu} + \frac{1}{1 + \mu} \right) A_{\mathcal{I}(x)} A_{\mathcal{I}(x)}^T. \tag{4.8}$$

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

4.2.1 Numerical Comparison

In this subsection, we compare our methods with two first-order solvers: ADMM [42] 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

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

Table 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 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 |

Table 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 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 |

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 (4.8) cannot be applied in ASSN.

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 numbers of A - and A^T -calls N_A and CPU time to compare the efficiency among different solvers.

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

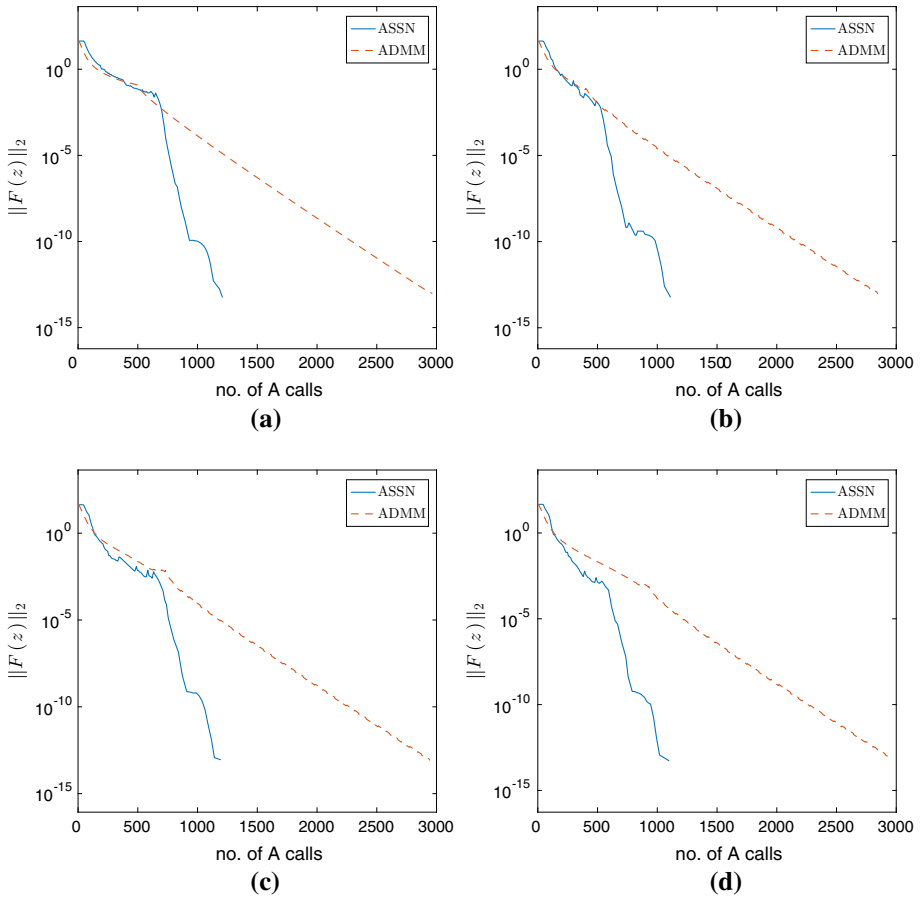


Fig. 3 Residual history with respect to the total numbers of A - and A^T -calls N_A . **a** 20 dB, **b** 40 dB, **c** 60 dB, **d** 80 dB

We also list the relative error so that we can compare ADMM, ASSN with SPGL1. These numerical results are reported in Tables 5, 6, 7 and 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 3 and 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 guar-

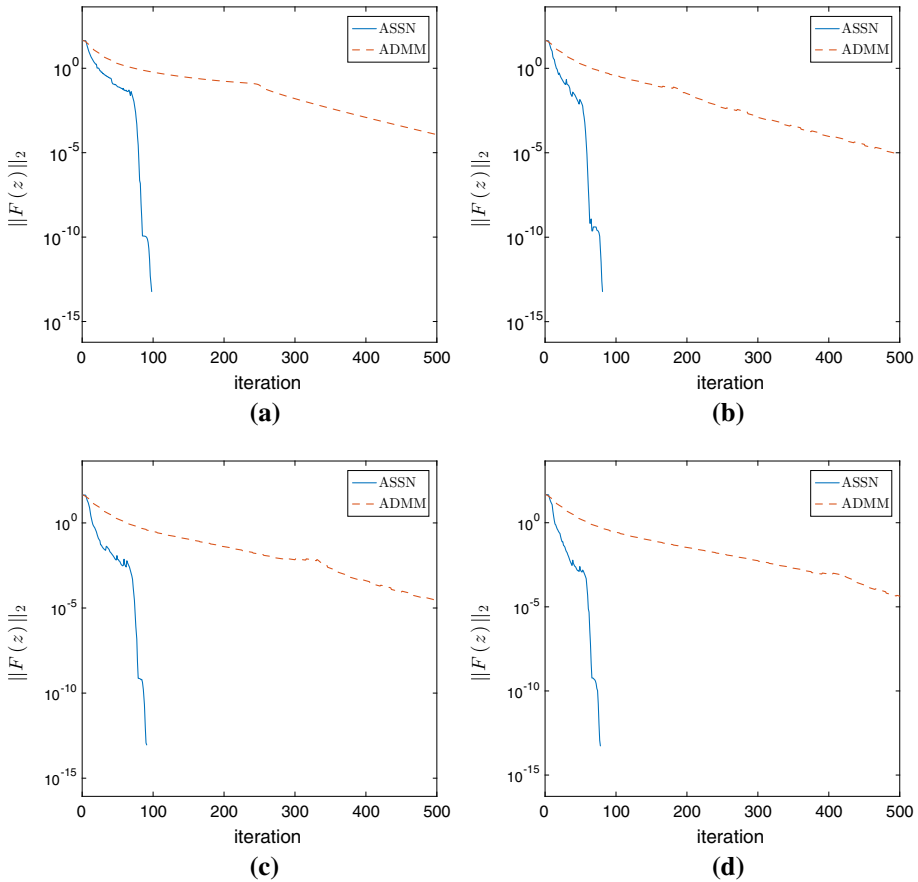


Fig. 4 Residual history with respect to the total numbers of iteration. **a** 20 dB, **b** 40 dB, **c** 60 dB, **d** 80 dB

anteed 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, although the BD-regular condition required for fast local convergence has not been established rigorously. The Jacobian of the ℓ_1 -regularized problem (4.1) is (4.3). The term $\partial^2 h(x)_{\mathcal{I}(x)\mathcal{I}(x)}$ behaves well in our numerical experiments. Similarly, the structure of ASA^T in (4.8) for the Jacobian of the basis-pursuit problem (4.4) also helps. Moreover, the number of the A - and A^T -calls of ASSN when solving the Newton systems shows that the CG method works well in general. It further indicates that the BD-regular condition may hold on these examples.

One of the key issues for solving the monotone nonsmooth equations is the convergence to global optimality. Note that, if $h(x)$ in problem (1.1) is an indicator function of a closed convex set C , the nonsmooth equations with respect to FBS are in the form of $F(z) = z - \Pi_C(x - \nabla f(x)) = 0$, which are equivalent to a variational inequality (VI) problem. Various

globally convergent projection-type or extragradient-type methods have been proposed to solve the VI, see the survey paper [41] and references therein. They may provide us plenty of alternative choices other than the hyperplane projection step in this paper.

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. In particular, the authors appreciate Dr. Andre Milzarek for reading the manuscript carefully and the help on improving the convergence analysis. The authors are grateful to the associate editor and two anonymous referees for their valuable comments and suggestions.

References

1. Ahookhosh, M., Amini, K., Bahrami, S.: Two derivative-free projection approaches for systems of large-scale nonlinear monotone equations. *Numer. Algorithms* **64**, 21–42 (2013)
2. Bauschke, H.H., Combettes, P.L.: *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, New York (2011)
3. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**, 1–122 (2011)
4. Byrd, R.H., Chin, G.M., Nocedal, J., Oztoprak, F.: A family of second-order methods for convex ℓ_1 -regularized optimization. *Math. Program.* **159**, 435–467 (2016)
5. Chen, C., Liu, Y.J., Sun, D., Toh, K.C.: A semismooth Newton-CG based dual PPA for matrix spectral norm approximation problems. *Math. Program.* **155**, 435–470 (2016)
6. Combettes, P.L., Pesquet, J.-C.: Proximal splitting methods in signal processing. In: Bauschke, H.H., Burachik, R., Combettes, P.L., Elser, V., Luke, D.R., Wolkowicz, H. (eds.) *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Volume 49 of Springer Optim. Appl., pp. 185–212. Springer, New York (2011)
7. Davis, D., Yin, W.: Convergence rate analysis of several splitting schemes. In: Glowinski, R., Osher, S.J., Yin, W. (eds.) *Splitting Methods in Communication, Imaging, Science, and Engineering*, Sci. Comput., pp. 115–163. Springer, Cham (2016)
8. Davis, D., Yin, W.: Faster convergence rates of relaxed Peaceman–Rachford and ADMM under regularity assumptions. *Math. Oper. Res.* **42**, 783–805 (2017)
9. De Luca, T., Facchinei, F., Kanzow, C.: A semismooth equation approach to the solution of nonlinear complementarity problems. *Math. Program.* **75**, 407–439 (1996)
10. Douglas, J., Rachford, H.H.: On the numerical solution of heat conduction problems in two and three space variables. *Trans. Am. Math. Soc.* **82**, 421–439 (1956)
11. Eckstein, J., Bertsekas, D.P.: On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.* **55**, 293–318 (1992)
12. Facchinei, F., Pang, J.-S.: *Finite-Dimensional Variational Inequalities and Complementarity Problems*, vol. II. Springer, New York (2003)
13. Fan, J.Y., Yuan, Y.X.: On the quadratic convergence of the Levenberg–Marquardt method without nonsingularity assumption. *Computing* **74**, 23–39 (2005)
14. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comput. Math. Appl.* **2**, 17–40 (1976)
15. Glowinski, R., Marrocco, A.: 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**, 41–76 (1975)
16. Griesse, R., Lorenz, D.A.: A semismooth Newton method for Tikhonov functionals with sparsity constraints. *Inverse Probl.* **24**, 035007 (2008)
17. Jiang, H., Qi, L.: Local uniqueness and convergence of iterative methods for nonsmooth variational inequalities. *J. Math. Anal. Appl.* **196**, 314–331 (1995)

18. Lee, J.D., Sun, Y., Saunders, M.A.: Proximal Newton-type methods for minimizing composite functions. *SIAM J. Optim.* **24**, 1420–1443 (2014)
19. Li, Q., Li, D.H.: A class of derivative-free methods for large-scale nonlinear monotone equations. *IMA J. Numer. Anal.* **31**, 1625–1635 (2011)
20. Lions, P.-L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.* **16**, 964–979 (1979)
21. Mifflin, R.: Semismooth and semiconvex functions in constrained optimization. *SIAM J. Control Optim.* **15**, 959–972 (1977)
22. Milzarek, A., Ulbrich, M.: A semismooth Newton method with multidimensional filter globalization for l_1 -optimization. *SIAM J. Optim.* **24**, 298–333 (2014)
23. Nocedal, J., Wright, S.J.: *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, 2nd edn. Springer, New York (2006)
24. Ortega, J.M., Rheinboldt, W.C.: *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York (1970)
25. Pang, J.-S., Qi, L.Q.: Nonsmooth equations: motivation and algorithms. *SIAM J. Optim.* **3**, 443–465 (1993)
26. Patrinos, P., Stella, L., Bemporad, A.: Forward–backward truncated Newton methods for convex composite optimization. <http://arxiv.org/abs/1402.6655>, 02 (2014)
27. Qi, L.Q.: Convergence analysis of some algorithms for solving nonsmooth equations. *Math. Oper. Res.* **18**, 227–244 (1993)
28. Qi, L.Q., Sun, J.: A nonsmooth version of Newton’s method. *Math. Program.* **58**, 353–367 (1993)
29. Rockafellar, R.T., Wets, R.J.-B.: *Variational Analysis*. Springer, Berlin (1998)
30. Scholtes, S.: *Introduction to Piecewise Differentiable Equations*, Springer Briefs in Optimization. Springer, New York (2012)
31. Shapiro, A.: Directionally nondifferentiable metric projection. *J. Optim. Theory Appl.* **81**, 203–204 (1994)
32. Solodov, M.V., Svaiter, B.F.: A globally convergent inexact Newton method for systems of monotone equations. In: Fukushima, M., Qi, L. (eds.) *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods* (Lausanne, 1997), vol. 22, pp. 355–369. Kluwer Academic Publishers, Dordrecht (1999)
33. Solodov, M.V., Svaiter, B.F.: A hybrid projection-proximal point algorithm. *J. Convex Anal.* **6**, 59–70 (1999)
34. Sun, D., Han, J.: Newton and quasi-Newton methods for a class of nonsmooth equations and related problems. *SIAM J. Optim.* **7**, 463–480 (1997)
35. Sun, D., Sun, J.: Semismooth matrix-valued functions. *Math. Oper. Res.* **27**, 150–169 (2002)
36. Tseng, P., Yun, S.: A coordinate gradient descent method for nonsmooth separable minimization. *Math. Program.* **117**, 387–423 (2009)
37. Ulbrich, M.: *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 (2011)
38. van den Berg, E., Friedlander, M.P.: Probing the Pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.* **31**, 890–912 (2008)
39. Wen, Z., Yin, W., Goldfarb, D., Zhang, Y.: A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation. *SIAM J. Sci. Comput.* **32**, 1832–1857 (2010)
40. Wright, S.J., Nowak, R.D., Figueiredo, M.A.T.: Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.* **57**, 2479–2493 (2009)
41. Xiu, N., Zhang, J.: Some recent advances in projection-type methods for variational inequalities. *J. Comput. Appl. Math.* **152**, 559–585 (2003)
42. Yang, J., Zhang, Y.: Alternating direction algorithms for ℓ_1 -problems in compressive sensing. *SIAM J. Sci. Comput.* **33**, 250–278 (2011)
43. Zhao, X.Y., Sun, D., Toh, K.C.: A Newton–CG augmented Lagrangian method for semidefinite programming. *SIAM J. Optim.* **20**, 1737–1765 (2010)
44. Zhao, Y.-B., Li, D.: Monotonicity of fixed point and normal mappings associated with variational inequality and its application. *SIAM J. Optim.* **11**, 962–973 (2001)
45. Zhou, G., Toh, K.C.: Superlinear convergence of a Newton-type algorithm for monotone equations. *J. Optim. Theory Appl.* **125**, 205–221 (2005)
46. Zhou, W.J., Li, D.H.: A globally convergent BFGS method for nonlinear monotone equations without any merit functions. *Math. Comput.* **77**, 2231–2240 (2008)