A primal-dual fixed point algorithm for convex separable minimization with applications to image restoration

Peijun Chen[1,2], Jianguo Huang[1,3], Xiaoqun Zhang[1,4]

[1] *Department of Mathematics, Shanghai Jiao Tong University, Shanghai 200240, China*

[2] *Department of Mathematics, Taiyuan University of Science and Technology, Taiyuan 030024, China*

[3] *Division of Computational Science, E-Institute of Shanghai Universities, Shanghai Normal University, China*

[4] *Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai 200240, China*

Email: chenpeijun@sjtu.edu.cn, jghuang@sjtu.edu.cn and xqzhang@sjtu.edu.cn

Abstract

Recently, minimization of a sum of two convex functions has received considerable interests in variational image restoration model. In this paper, we propose a general algorithmic framework for solving separable convex minimization problem from the point of view of fixed point algorithms based on proximity operators [23]. Motivated from proximal forward-backward splitting (PFBS) proposed in [13] and fixed point algorithms based on the proximity operator (FP^{2}O) for image denoising [21], we design a primal-dual fixed point algorithm based on proximity operator (PDFP^{2}O$_\kappa$ for $\kappa \in [0, 1)$) and obtain a scheme with close form for each iteration. Using the firmly nonexpansive properties of the proximity operator and with the help of a special norm over a product space, we achieve the convergence of the proposed PDFP^{2}O$_\kappa$ algorithm. Moreover, under some stronger assumptions, we can prove the global linear convergence of the proposed algorithm. We also give the connection of the proposed algorithm with other existing first order methods. Finally, we illustrate the efficiency of PDFP^{2}O$_\kappa$ through some numerical examples on image supperresolution and computerized tomographic reconstruction.

Keywords: proximity operator, fixed point algorithm, primal-dual form, image restoration.

1 Introduction

This paper is devoted to designing and discussing an efficient algorithmic framework for minimizing the sum of two proper lower semicontinuous convex functions, i.e.

$$x^* = \arg\min_{x\in\mathbb{R}^n} \quad (f_1 \circ B)(x) + f_2(x), \tag{1.1}$$

where $f_1 \in \Gamma_0(\mathbb{R}^m)$, $f_2 \in \Gamma_0(\mathbb{R}^n)$ and f_2 is differentiable on $\mathbb{R}^n$ with a $1/\beta$-Lipschitz continuous gradient for some $\beta \in (0, +\infty)$ and $B : \mathbb{R}^n \to \mathbb{R}^m$ a linear transform. Here and in what follows, for a real Hilbert

space $\mathcal{X}$, $\Gamma_0(\mathcal{X})$ denotes the collection of all proper lower semicontinuous convex functions from $\mathcal{X}$ to $(-\infty, +\infty]$. Despite its simplicity, many problems in image processing can be formulated in the form of (1.1). For instance, the following variational sparse recovery models are often considered in image restoration and medical image reconstruction:

$$x^* = \arg\min_{x\in\mathbb{R}^n} \quad \mu\|Bx\|_1 + \frac{1}{2}\|Ax - b\|_2^2, \tag{1.2}$$

where $\|\cdot\|_2$ denotes the usual Euclidean norm for a vector, A is a $p \times n$ matrix representing a linear transform, $b \in \mathbb{R}^p$ and $\mu > 0$ is the regularization parameter. The term$\|Bx\|_1$ is the usual ℓ^1 based regularization in order to promote sparsity under the transform B. For example, for the well-known Rudin-Osher-Fatemi model (ROF) [30] $\|Bx\|_1$ represents the total variation seminorm which aims to recover piecewise constant images, with B a $2n \times n$ discrete differential matrix (cf. [16, 21]). More precisely, $\|Bx\|_1$ and $\|Bx\|_{1,2}$ are for anisotropic total-variation and isotropic total-variation, respectively, and here we simply write them as $\|Bx\|_1$. The problem (1.2) can be expressed in the form of (1.1) by setting $f_1 = \mu\|\cdot\|_1$ and $f_2(x) = \frac{1}{2}\|Ax - b\|_2^2$. One of the main difficulty in solving it is that f_1 is nondifferentiable. The case often occurs in many problems we are interested.

Another general problem often considered in the literature takes the following form:

$$x^* = \arg\min_{x\in\mathcal{X}} \quad f(x) + h(x), \tag{1.3}$$

where $f,\ h \in \Gamma_0(\mathcal{X})$ and h is differentiable on $\mathcal{X}$ with a $1/\beta$-Lipschitz continuous gradient for some $\beta \in (0, +\infty)$. The problem (1.1) that we are interested in this paper can be viewed as a special case of problem (1.3) for $\mathcal{X} = \mathbb{R}^n$ and $f = f_1 \circ B$, $h = f_2$. On the other hand, we can also consider that the problem (1.3) is a special case of problem (1.1) for $\mathcal{X} = \mathbb{R}^n$, $f_2 = h, f_1 = f$ and $B = I$, where I denotes the usual identity operator. For problem (1.3), Combettes and Wajs proposed in [13] a proximal forward backward splitting (PFBS) algorithm, i.e.

$$x_{k+1} = \mathrm{prox}_{\gamma f}(x_k - \gamma\nabla h(x_k)), \tag{1.4}$$

where $0 < \gamma < 2\beta$ is a stepsize parameter, and the operator prox_f is defined by

$$\begin{array}{rccl} \mathrm{prox}_f : & \mathcal{X} & \to & \mathcal{X} \\ & x & \mapsto & \arg\min\limits_{y\in\mathcal{X}} f(y) + \frac{1}{2}\|x - y\|_2^2, \end{array}$$

called the proximity operator of f. Note that this type of splitting method was originally studied in [20, 28] for solving partial differential equations, and the notion of proximity operators was first introduced by Moreau in [23] as a generalization of projection operators. The iteration (1.4) consists of two sequential steps. First one performs a forward (explicit) step involving the evaluation of the gradient of h; then one performs a backward (implicit) step involving f. This numerical scheme is very simple and efficient when the proximity operator used in the second step can be carried out efficiently. For example when $f = \|\cdot\|_1$ for sparse regularization, the proximity operator $\mathrm{prox}_{\gamma f}(x)$ can be written as the famous component-wise soft-thresholding (also known as a shrinkage) operation. However, the proximity operators for the general form $f = f_1 \circ B$ as in (1.1) do not have an explicit expression, leading to numerical solution of a difficult subproblem. In fact, the subproblem of (1.2) is $\mathrm{prox}_{\mu\|\cdot\|_1\circ B}(b)$ and often formulated as the ROF denoising problem:

$$x^* = \arg\min_{x\in\mathbb{R}^n} \quad \mu\|Bx\|_1 + \frac{1}{2}\|x - b\|_2^2,$$

where $b \in \mathbb{R}^n$ denotes a corrupted image to be denoised.

In recent years, many splitting methods have been designed to solve the last subproblem in order to take advantage of the efficiency of soft-thresholding operator. For example, Goldstein-Osher proposed in [18] a splitting algorithm based on Bregman iteration, namely split Bregman, to implement the action of $\text{prox}_{f_1 \circ B}$, in particular for total variation minimization. This algorithm has shown to be very efficient and useful for a large class of convex separable programming problems. Theoretically, it is shown to be equivalent to Douglas-Rachford splitting algorithm (DRS, see [31, 14]) and alternating direction of multiplier method (ADMM, see [15, 5]), and the convergence was then analyzed based on such equivalence. The split Bregman proposed in [18] is also designed to solve the convex separable problem (1.1). In particular, for the variational model (1.2), the subproblem involves solving a quadratic minimization, which sometimes can be time consuming. To overcome this, a primal-dual inexact split Uzawa methods was proposed in [37] to maximally decouple the subproblems so that each iteration step is precise and explicit. In [16, 9], more theoretical analysis on the variants of the primal-dual type method and the connection with existing methods were examined to bridge the gap between different types of methods. Also, the convergence of ADMM was further analyzed in [19] based on proximal point algorithm (PPA) formulation.

In this paper, we will follow a different point of view. In [21], Micchelli-Shen-Xu designed an algorithm called (FP^{2}O) to solve $\text{prox}_{f_1 \circ B}(x)$. We aim to extend FP^{2}O to solve the general problem (1.1) with maximally decoupled iteration scheme. One obvious advantage of the proposed scheme is very easy for parallel implementation. Furthermore, we will show that the proposed algorithm is convergent in general setting. Under some assumptions of the convex function f_2 and the linear transform B, we can further prove the linear convergence rate of the method under the framework of fixed point iteration. Note that most of the existing works based on ADMM have shown a sub-linear convergence rate $O(1/k)$ on the objective function and $O(1/k^2)$ on the accelerated version, where k is the iteration number. Recently in [19], the ergodic and non-ergodic convergence on the difference of two sequential primal-dual sequences were analyzed. In this paper, we will prove the convergence rate of the iterations directly from the point of view of fixed point theory under some common assumptions. During the preparation of this paper, we notice that Yin and Peng [35] also considered the global linear convergence of the ADMM and its variants based on similar assumptions. Furthermore, we will reformulate the fixed point type of methods and show the connections with some existing first order methods for (1.1) and (1.2) for better understanding.

The rest of the paper is organized as follows. In Section 2, we recall the fixed point algorithm FP^{2}O and some related works and then deduce the proposed PDFP^{2}O algorithm and its extension PDFP^{2}O$_\kappa$ from our intuitions. In Section 3, we first deduce PDFP^{2}O$_\kappa$ again in the setting of fixed point iteration; we then establish its convergence under general setting and the convergence rate under some stronger assumptions on f_2 and B. In Section 4, we give the equivalent form of PDFP^{2}O, the relationships and differences with other first order algorithms. In the final section, we will show the numerical performance and efficiency of PDFP^{2}O$_\kappa$ through image superresolution and CT image reconstruction examples.

2 Fixed point algorithms

Similar to the fixed point algorithm on the dual for ROF denoising model proposed by Chambolle [8], Micchelli et al. proposed an algorithm called (FP^{2}O) in [21] to solve the proximity operator $\text{prox}_{f_1 \circ B}(b)$ for $b \in \mathbb{R}^n$, especially for total variation based image denoising. Let $\lambda_{max}(BB^T)$ be the largest eigenvalue of BB^T. For $0 < \lambda < 2/\lambda_{max}(BB^T)$, we define the operator

$$H(v) = (I - \text{prox}_{\frac{f_1}{\lambda}})(Bb + (I - \lambda BB^T)v) \quad \text{for all } v \in \mathbb{R}^m, \tag{2.1}$$

then FP^{2}O algorithm is described as **Algorithm 1**, where H_κ is the κ-averaged operator of H, i.e. $H_\kappa = \kappa I + (1-\kappa)H$ for $\kappa \in (0,1)$, see definition 3.3 in the next section.

Algorithm 1 Fixed point algorithm based on proximity operator, FP^{2}O [21].

Step 1: Set $v_0 \in \mathbb{R}^m$, $0 < \lambda < 2/\lambda_{max}(BB^T)$, $\kappa \in (0,1)$.
Step 2: Calculate v^*, which is the fixed point of H, with iteration $v_{k+1} = H_\kappa(v_k)$.
Step 3: $\text{prox}_{f_1 \circ B}(b) = b - \lambda B^T v^*$.

The key technique to obtain FP^{2}O scheme relies on the relation of the subdifferential of a convex function and its proximity operator, as described in the result (3.1). An advantage of FP^{2}O is that its iteration does not require subproblem solving and the convergence is analyzed in the classical framework of fixed point iteration. This algorithm has been extended in [1, 10] to solve

$$x^* = \arg\min_{x \in \mathbb{R}^n} \quad (f_1 \circ B)(x) + \frac{1}{2} x^T Q x - b^T x,$$

where $Q \in \mathrm{M}_n$, with M_n the collection of all symmetric positive definite $n \times n$ matrices, $b \in \mathbb{R}^n$. Define

$$\widetilde{H}(v) = (I - \text{prox}_{\frac{f_1}{\lambda}})(BQ^{-1}b + (I - \lambda BQ^{-1}B^T)v) \ \text{ for all } v \in \mathbb{R}^m.$$

Then, the corresponding algorithm is given below, called **Algorithm 2**, which can be viewed as a fixed point algorithm based on inverse matrix and proximity operator or FP^{2}O based on inverse matrix (IFP^{2}O). Here the matrix Q is assumed to be invertible and the inverse can be easily calculated, which is unfortunately not the case in most of applications in imaging science. Moreover, there is no theoretical guarantee of convergence if the linear system is only solved approximately.

Algorithm 2 FP^{2}O based on inverse matrix, IFP^{2}O [1].

Step 1: Set $v_0 \in \mathbb{R}^m$ and $0 < \lambda < 2/\lambda_{max}(BQ^{-1}B^T)$, $\kappa \in (0,1)$.
Step 2: Calculate $\widetilde{v}^*$, which is the fixed point of $\widetilde{H}$, with iteration $\widetilde{v}_{k+1} = \widetilde{H}_\kappa(\widetilde{v}_k)$.
Step 3: $x^* = Q^{-1}(b - \lambda B^T \widetilde{v}^*)$.

Further, the authors in [1] combined PFBS and FP^{2}O for solving problem (1.1), for which we call proximal forward-backward splitting based on FP^{2}O (PFBS_FP^{2}O), see **Algorithm 3**. The acceleration combing with Nesterov' method [24, 32, 33] was also considered in [1].

Algorithm 3 Proximal forward-backward splitting based on FP^{2}O, PFBS_FP^{2}O [1].

Step 1: Set $x_0 \in \mathbb{R}^n$, $0 < \gamma < 2\beta$.
Step 2: for $k = 0, 1, 2, \cdots$
Calculate $x_{k+1} = \text{prox}_{\gamma f_1 \circ B}(x_k - \gamma \nabla f_2(x_k))$ using FP^{2}O (see **Algorithm 1**)
end for

At step k of **Algorithm 3**, after one forward iteration $x_{k+1/2} = x_k - \gamma \nabla f_2(x_k)$, we need to solve $\text{prox}_{\gamma f_1 \circ B}(x_{k+1/2})$. So **Algorithm 3** involves inner and outer iterations, and it is often problematic to set the appropriate inner stopping conditions to balance computational time and precision. We intend to propose an algorithm which does not involve any inner iterations. Instead of using many number of inner fixed point iterations for solving $\text{prox}_{f_1 \circ B}(x)$, we can use only one inner fixed point iteration. For this, define

$$H_{k+1}(v) = (I - \text{prox}_{\frac{\gamma}{\lambda} f_1})(Bx_{k+1/2} + (I - \lambda BB^T)v) \ \text{ for all } v \in \mathbb{R}^m.$$

Suppose $\kappa = 0$ in FP^{2}O. A very natural idea is to take the numerical solution v_k for H_k as the initial value, and only do one iteration for solving the fixed point of H_{k+1}, then we can get the following iteration scheme:

$$(\text{PDFP}^2\text{O}) \quad \begin{cases} v_{k+1} = (I - \text{prox}_{\frac{\gamma}{\lambda} f_1})(B(x_k - \gamma \nabla f_2(x_k)) + (I - \lambda B B^T) v_k), & (2.2a) \\ x_{k+1} = x_k - \gamma \nabla f_2(x_k) - \lambda B^T v_{k+1}, & (2.2b) \end{cases}$$

which produces our proposed method **Algorithm 4**, described below. This algorithm can also be deduced from the fixed point formulation, which we will give the detail in the next section. On the other hand, since x is the primal variable related to (1.1), it is very natural to ask what role the variable v plays in our algorithm. After a thorough study, we find out as given in Section 4.1 that v is actually the dual variable of the primal-dual form related to (1.1). Based on these observations, we call our method a primal-dual fixed point algorithm based on proximity operator, and abbreviate it as PDFP^{2}O inheriting the notion of "FP^{2}O". If $B = I$, $\lambda = 1$, then form (2.2) is equivalent to form (1.4). So PFPS can be seen as a special case of PDFP^{2}O. Also, when $f_2(x) = \frac{1}{2}\|x - b\|_2^2$ and $\gamma = 1$, then PDFP^{2}O reduces to FP^{2}O for solving $\text{prox}_{f_1 \circ B}(b)$ with $\kappa = 0$. For general B and f_2, each step of the proposed algorithm is explicit when $\text{prox}_{\frac{\gamma}{\lambda} f_1}$ is easy to compute. Note that the technique of approximating the subproblem by only one iteration is also proposed in a primal-dual inexact Uzawa framework in [37]. We will show the connection to this algorithm and other ones in Section 4.

Algorithm 4 Primal-dual fixed points algorithm based on proximity operator, PDFP^{2}O.

Step 1: Set $x_0 \in \mathbb{R}^n$, $v_0 \in \mathbb{R}^m$, $0 < \lambda \leq 1/\lambda_{max}(BB^T)$, $0 < \gamma < 2\beta$.

Step 2: for $k = 0, 1, 2, \cdots$

$$\begin{aligned} x_{k+1/2} &= x_k - \gamma \nabla f_2(x_k), \\ v_{k+1} &= (I - \text{prox}_{\frac{\gamma}{\lambda} f_1})(B x_{k+1/2} + (I - \lambda B B^T) v_k), \\ x_{k+1} &= x_{k+1/2} - \lambda B^T v_{k+1}. \end{aligned}$$

end for

Borrowing the fixed point formulation of PDFP^{2}O, we can introduce a relaxation parameter $\kappa \in [0, 1)$ to get **Algorithm 5**, which is exactly a Picard method with parameters. If $\kappa = 0$, PDFP^{2}O$_\kappa$ reduces to PDFP^{2}O. Our theoretical analysis for PDFP^{2}O$_\kappa$ given in the next section is mainly based on this fixed point setting.

Algorithm 5 PDFP^{2}O$_\kappa$.

Step 1: Set $x_0 \in \mathbb{R}^n$, $v_0 \in \mathbb{R}^m$, $0 < \lambda \leq 1/\lambda_{max}(BB^T)$, $0 < \gamma < 2\beta$, $\kappa \in [0, 1)$.

Step 2: for $k = 0, 1, 2, \cdots$

$$\begin{aligned} x_{k+1/2} &= x_k - \gamma \nabla f_2(x_k), \\ \widetilde{v}_{k+1} &= H_{k+1}(v_k), \\ \widetilde{x}_{k+1} &= x_{k+1/2} - \lambda B^T \widetilde{v}_{k+1}, \\ v_{k+1} &= \kappa v_k + (1 - \kappa)\widetilde{v}_{k+1}, \\ x_{k+1} &= \kappa x_k + (1 - \kappa)\widetilde{x}_{k+1}. \end{aligned}$$

end for

3 Convergence analysis

3.1 General convergence

First of all, let us mention some related definitions and lemmas for later requirements. From now on, we use $\mathcal{X}$ to denote a finite-dimensional real Hilbert space. Moreover, we always assume that problem (1.1) has at least one solution. As shown in [13], if the objective function $(f_1 \circ B)(x) + f_2(x)$ is coercive, i.e.

$$\lim_{\|x\| \to +\infty} ((f_1 \circ B)(x) + f_2(x)) = +\infty,$$

then the existence of solution can be ensured for (1.1).

Definition 3.1 (Subdifferential [29]) *Let f be a function in $\Gamma_0(\mathcal{X})$. The subdifferential of f is the set-valued operator $\partial f : \mathcal{X} \to 2^{\mathcal{X}}$, the value of which at $x \in \mathcal{X}$ is*

$$\partial f(x) = \{v \in \mathcal{X} \mid \langle v, y - x\rangle + f(x) \leq f(y) \ \ \text{for all } y \in \mathcal{X}\},$$

where $\langle \cdot, \cdot \rangle$ denotes the inner-product over $\mathcal{X}$.

Definition 3.2 (Nonexpansive operators and firmly nonexpansive operators [29]) *An operator $T : \mathcal{X} \to \mathcal{X}$ is nonexpansive if and only if it satisfies:*

$$\|Tx - Ty\|_2 \leq \|x - y\|_2 \ \ \text{for all } (x, y) \in \mathcal{X}^2.$$

T is firmly nonexpansive if and only if it satisfies one of the following equivalent conditions:
(i) $\|Tx - Ty\|_2^2 \leq \langle Tx - Ty, x - y\rangle$ *for all* $(x, y) \in \mathcal{X}^2$.
(ii) $\|Tx - Ty\|_2^2 \leq \|x - y\|_2^2 - \|(I - T)x - (I - T)y\|_2^2$ *for all* $(x, y) \in \mathcal{X}^2$.

It is easy to show from the above definitions that a firmly nonexpansive operator T is nonexpansive.

Definition 3.3 (Picard sequence, κ-averaged operator [25]) *Let $T : \mathcal{X} \to \mathcal{X}$ be an operator. For a given initial point $u_0 \in \mathcal{X}$, the Picard sequence of the operator T is defined by $u_{k+1} = T(u_k)$, for $k \in \mathbb{N}$. For a real number $\kappa \in (0, 1)$, the κ-averaged operator T_κ of T is defined by $T_\kappa = \kappa I + (1 - \kappa)T$. We also write $T_0 = T$.*

Lemma 3.1 *Suppose $f \in \Gamma_0(\mathbb{R}^m)$ and $x \in \mathbb{R}^m$. Then there holds*

$$y \in \partial f(x) \Leftrightarrow x = \text{prox}_f(x + y). \tag{3.1}$$

Furthermore, if f has $1/\beta$-Lipschitz continuous gradient, then

$$\langle \nabla f(x) - \nabla f(y), x - y\rangle \geq \beta \|\nabla f(x) - \nabla f(y)\|^2 \ \ \text{for all } (x, y) \in \mathcal{X}^2. \tag{3.2}$$

Proof. The first result is nothing but Proposition 2.6 of [21]. If f has $1/\beta$-Lipschitz continuous gradient, we have from [13] that $\beta \nabla f$ is firmly nonexpansive, which implies (3.2) readily. ■

Lemma 3.2 (Lemma 2.4 of [13]) *Let f be a function in $\Gamma_0(\mathbb{R}^m)$. Then prox_f and $I - \text{prox}_f$ are both firmly nonexpansive operators.*

Lemma 3.3 (Opial κ-averaged theorem, Theorem 3 of [25]) *If S is a closed and convex set in $\mathcal{X}$ and $T : S \to S$ is a nonexpansive mapping having at least one fixed point, then for $\kappa \in (0, 1)$, T_κ is nonexpansive, maps S to itself, and has the same set of the fixed points as T. Furthermore, for any $u_0 \in S$ and $\kappa \in (0, 1)$, the Picard sequence of T_κ converges to a fixed point of T.*

Now, we are ready to get a fixed point formulation for the solution of problem (1.1) and discuss the convergence of PDFP^{2}O$_\kappa$. To this end, for any two positive numbers λ and γ, define $T_1 : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$ as

$$T_1(v,x) = (I - \mathrm{prox}_{\frac{\gamma}{\lambda} f_1})(B(x - \gamma \nabla f_2(x)) + (I - \lambda BB^T)v) \tag{3.3}$$

and $T_2 : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$ as

$$T_2(v,x) = x - \gamma \nabla f_2(x) - \lambda B^T \circ T_1. \tag{3.4}$$

Denote $T : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m \times \mathbb{R}^n$ as

$$T(v,x) = \left(T_1(v,x),\ T_2(v,x)\right). \tag{3.5}$$

Theorem 3.1 *Let λ and γ be two positive numbers. Suppose that x^* is a solution of (1.1). Then there exists $v^* \in \mathbb{R}^m$ such that*

$$\begin{cases} v^* = T_1(v^*, x^*), \\ x^* = T_2(v^*, x^*). \end{cases}$$

In other words, $u^* = (v^*, x^*)$ is a fixed point of T. Conversely, if $u^* \in \mathbb{R}^m \times \mathbb{R}^n$ is a fixed point of T, with $u^* = (v^*, x^*)$, $v^* \in \mathbb{R}^m$, $x^* \in \mathbb{R}^n$, then x^* is a solution of (1.1).

Proof. By the first order optimality condition of the problem (1.1), we have

$$\begin{aligned} & x^* = \arg\min_{x \in \mathbb{R}^n} \quad (f_1 \circ B)(x) + f_2(x) \\ \Leftrightarrow & 0 \in -\nabla f_2(x^*) - \partial(f_1 \circ B)(x^*) \\ \Leftrightarrow & 0 \in -\gamma \nabla f_2(x^*) - \gamma \partial(f_1 \circ B)(x^*) \\ \Leftrightarrow & x^* \in x^* - \gamma \nabla f_2(x^*) - \lambda(B^T \circ \frac{\gamma}{\lambda} \partial f_1 \circ B)(x^*). \end{aligned}$$

Let

$$v^* \in (\frac{\gamma}{\lambda} \partial f_1 \circ B)(x^*) = \partial(\frac{\gamma}{\lambda} f_1)(Bx^*). \tag{3.7}$$

Then

$$x^* = x^* - \gamma \nabla f_2(x^*) - \lambda B^T v^*. \tag{3.8}$$

Moreover, it follows from result (3.1) that (3.7) is equivalent to

$$\begin{aligned} & Bx^* = \mathrm{prox}_{\frac{\gamma}{\lambda} f_1}(Bx^* + v^*) \\ \Leftrightarrow & (Bx^* + v^*) - v^* = \mathrm{prox}_{\frac{\gamma}{\lambda} f_1}(Bx^* + v^*) \\ \Leftrightarrow & v^* = (I - \mathrm{prox}_{\frac{\gamma}{\lambda} f_1})(Bx^* + v^*). \end{aligned} \tag{3.9}$$

Inserting (3.8) into (3.9) gives

$$v^* = (I - \mathrm{prox}_{\frac{\gamma}{\lambda} f_1})(B(x^* - \gamma \nabla f_2(x^*)) + (I - \lambda BB^T)v^*).$$

This shows $v^* = T_1(v^*, x^*)$. Next, replacing v^* in (3.8) by $T_1(v^*, x^*)$, we readily have $x^* = T_2(v^*, x^*)$. Therefore, for $u^* = (v^*, x^*)$, $u^* = T(u^*)$.

On the other hand, if $u^* = T(u^*)$, we can derive that x^* satisfies the first order optimality condition of (1.1). Therefore we conclude that x^* is a minimizer of (1.1). ■

In the following, we will show the algorithm PDFP^{2}O$_\kappa$ is a Picard method related to the operator T_κ.

Theorem 3.2 *Suppose $\kappa \in [0,1)$. Set $T_\kappa = \kappa I + (1-\kappa)T$. Then the Picard sequence $\{u_k\}$ of T_κ is exactly the one obtained by algorithm PDFP^{2}O$_\kappa$.*

Proof. According to the definitions in (3.3)-(3.5), the component form of $u_{k+1} = T(u_k)$ can be expressed as

$$\begin{cases} v_{k+1} = T_1(v_k, x_k) = (I - \text{prox}_{\frac{\gamma}{\lambda} f_1})(B(x_k - \gamma\nabla f_2(x_k)) + (I - \lambda BB^T)v_k) \\ x_{k+1} = T_2(v_k, x_k) = x_k - \gamma\nabla f_2(x_k) - \lambda B^T \circ T_1(v_k, x_k) = x_k - \gamma\nabla f_2(x_k) - \lambda B^T v_{k+1}. \end{cases}$$

Therefore, the iteration $u_{k+1} = T(u_k)$ is equivalent to (2.2). Employing the similar argument we can get the conclusion for general T_κ with $\kappa \in [0,1)$. ■

Remark 3.1 *From the last result, we find out that algorithm PDFP^{2}O$_\kappa$ can also be obtained in the setting of fixed point iteration immediately.*

For the convergence analysis for PDFP^{2}O$_\kappa$, we will first prove a key inequality for general cases (cf. equation (3.14)). Denote

$$g(x) = x - \gamma\nabla f_2(x) \quad \text{for all } x \in \mathbb{R}^n, \tag{3.10}$$

$$M = I - \lambda BB^T. \tag{3.11}$$

When $0 < \lambda \leq 1/\lambda_{\max}(BB^T)$, M is a symmetric positive semi-definite matrix, so we can define semi-norm:

$$\|v\|_M = \sqrt{\langle v, Mv\rangle} \quad \text{for all } v \in \mathbb{R}^m. \tag{3.12}$$

For an element $u = (v, x) \in \mathbb{R}^m \times \mathbb{R}^n$, with $v \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$, let

$$\|u\|_\lambda = \sqrt{\|x\|_2^2 + \lambda\|v\|_2^2}. \tag{3.13}$$

We can easily see that $\|\cdot\|_\lambda$ is a norm over the produce space $\mathbb{R}^m \times \mathbb{R}^n$ whenever $\lambda > 0$.

Lemma 3.4 *For any two elements $u_1 = (v_1, x_1)$, $u_2 = (v_2, x_2)$ in $\mathbb{R}^m \times \mathbb{R}^n$, there holds*

$$\begin{aligned} \|T(u_1) - T(u_2)\|_\lambda^2 \leq & \|u_1 - u_2\|_\lambda^2 - \gamma(2\beta - \gamma)\|\nabla f_2(x_1) - \nabla f_2(x_2)\|_2^2 \\ & - \|\lambda B^T(v_1 - v_2)\|_2^2 - \lambda\|(T_1(u_1) - T_1(u_2)) - (v_1 - v_2)\|_M^2. \end{aligned} \tag{3.14}$$

Proof. By Lemma 3.2, $I - \text{prox}_{\frac{\gamma}{\lambda} f_1}$ is a firmly nonexpensive operator. This together with (3.3), (3.10) and (3.11) yields

$$\begin{aligned} & \|T_1(u_1) - T_1(u_2)\|_2^2 \\ \leq & \langle T_1(u_1) - T_1(u_2), B(g(x_1) - g(x_2)) + M(v_1 - v_2)\rangle \\ = & \langle T_1(u_1) - T_1(u_2), B(g(x_1) - g(x_2))\rangle + \langle T_1(u_1) - T_1(u_2), M(v_1 - v_2)\rangle. \end{aligned} \tag{3.15}$$

It follows from (3.4), (3.10), (3.11) and (3.12) that

$$\begin{aligned} & \|T_2(u_1) - T_2(u_2)\|_2^2 \\ = & \|(g(x_1) - g(x_2)) - \lambda B^T \circ (T_1(u_1) - T_1(u_2))\|_2^2 \\ = & \|g(x_1) - g(x_2)\|_2^2 - 2\lambda\langle B^T \circ (T_1(u_1) - T_1(u_2)), g(x_1) - g(x_2)\rangle \\ & + \|\lambda B^T \circ (T_1(u_1) - T_1(u_2))\|_2^2 \\ = & \|g(x_1) - g(x_2)\|_2^2 - 2\lambda\langle T_1(u_1) - T_1(u_2), B(g(x_1) - g(x_2))\rangle \\ & - \lambda\|T_1(u_1) - T_1(u_2)\|_M^2 + \lambda\|T_1(u_1) - T_1(u_2)\|_2^2. \end{aligned} \tag{3.16}$$

Observing the definitions in (3.5) and (3.10)-(3.13), we have by (3.15)-(3.16) that

$$\begin{aligned}
&\|T(u_1) - T(u_2)\|_\lambda^2 \\
=&\|T_2(u_1) - T_2(u_2)\|_2^2 + \lambda\|T_1(u_1) - T_1(u_2)\|_2^2 \\
=&\|g(x_1) - g(x_2)\|_2^2 - 2\lambda\langle T_1(u_1) - T_1(u_2), B(g(x_1) - g(x_2))\rangle \\
&- \lambda\|T_1(u_1) - T_1(u_2)\|_M^2 + 2\lambda\|T_1(u_1) - T_1(u_2)\|_2^2 \\
\leq&\|g(x_1) - g(x_2)\|_2^2 - \lambda\|T_1(u_1) - T_1(u_2)\|_M^2 + 2\lambda\langle T_1(u_1) - T_1(u_2), M(v_1 - v_2)\rangle \\
=&\|g(x_1) - g(x_2)\|_2^2 + \lambda\|v_1 - v_2\|_M^2 - \lambda\|(T_1(u_1) - T_1(u_2)) - (v_1 - v_2)\|_M^2. \qquad (3.17)
\end{aligned}$$

Using the definition in (3.10) and estimate (3.2), we know

$$\begin{aligned}
&\|g(x_1) - g(x_2)\|_2^2 \\
=&\|x_1 - x_2\|_2^2 - 2\gamma\langle\nabla f_2(x_1) - \nabla f_2(x_2), x_1 - x_2\rangle + \gamma^2\|\nabla f_2(x_1) - \nabla f_2(x_2)\|_2^2 \\
\leq&\|x_1 - x_2\|_2^2 - \gamma(2\beta - \gamma)\|\nabla f_2(x_1) - \nabla f_2(x_2)\|_2^2. \qquad (3.18)
\end{aligned}$$

By the definitions in (3.11) and (3.12),

$$\lambda\|v_1 - v_2\|_M^2 = \lambda\|v_1 - v_2\|_2^2 - \|\lambda B^T(v_1 - v_2)\|_2^2. \qquad (3.19)$$

Recalling the definition in (3.13), we easily know (3.14) is a direct consequence of (3.17)-(3.19). ■

From Lemma 3.4, we can derive the following result.

Corollary 3.3 *If $0 < \gamma < 2\beta$, $0 < \lambda \leq 1/\lambda_{\max}(BB^T)$, then T is nonexpensive under the norm of $\|\cdot\|_\lambda$.*

Since T is nonexpensive, we are able to show the convergency of PDFP^{2}O$_\kappa$ for $\kappa \in (0, 1)$, in view of Lemma 3.3.

Theorem 3.4 *Suppose $0 < \gamma < 2\beta$, $0 < \lambda \leq 1/\lambda_{\max}(BB^T)$ and $\kappa \in (0, 1)$. Let $u_k = (v_k, x_k)$ be a sequence generated by PDFP^{2}O$_\kappa$. Then $\{u_k\}$ converges to a fixed point of T and $\{x_k\}$ converges to a solution of problem* (1.1).

Proof. In view of Theorem 3.2, we know $u_{k+1} = T_\kappa(u_k)$, so $\{u_k\}$ is the Picard sequence of T_κ. By assumption, problem (1.1) has a solution, and hence operator T has a fixed point from Theorem 3.1. According to Corollary 3.3, T is nonexpensive. Therefore, by letting $S = \mathbb{R}^m$, we find from Lemma 3.3 that $\{u_k\}$ converges to a fixed point of T for $\kappa \in (0, 1)$. With this result in mind, $\{x_k\}$ converges to a solution of problem (1.1) from Theorem 3.1. ■

Now, let us proceed with the convergence analysis of PDFP^{2}O using some novel technique.

Theorem 3.5 *Suppose $0 < \gamma < 2\beta$ and $0 < \lambda \leq 1/\lambda_{\max}(BB^T)$. Let $u_k = (v_k, x_k)$ be the sequence generated by PDFP^{2}O. Then the sequence $\{u_k\}$ converges to a fixed point of T, and the sequence $\{x_k\}$ converges to a solution of problem* (1.1).

Proof. Let $u^* = (v^*, x^*) \in \mathbb{R}^m \times \mathbb{R}^n$ be a fixed point of T. Using Lemma 3.4, we have

$$\begin{aligned}
\|u_{k+1} - u^*\|_\lambda^2 \leq&\|u_k - u^*\|_\lambda^2 - \gamma(2\beta - \gamma)\|\nabla f_2(x_k) - \nabla f_2(x^*)\|_2^2 \\
&- \|\lambda B^T(v_k - v^*)\|_2^2 - \lambda\|v_{k+1} - v_k\|_M^2. \qquad (3.20)
\end{aligned}$$

Summing (3.20) over k from 0 to $+\infty$ gives

$$\sum_{k=1}^{+\infty}\left\{\gamma(2\beta - \gamma)\|\nabla f_2(x_k) - \nabla f_2(x^*)\|_2^2 + \|\lambda B^T(v_k - v^*)\|_2^2 + \lambda\|v_{k+1} - v_k\|_M^2\right\} \leq \|u_1 - u^*\|_\lambda^2.$$

So

$$\lim_{k\to+\infty}\left\{\gamma(2\beta-\gamma)\|\nabla f_2(x_k)-\nabla f_2(x^*)\|_2^2+\|\lambda B^T(v_k-v^*)\|_2^2+\lambda\|v_{k+1}-v_k\|_M^2\right\}=0,$$

which together with $0<\gamma<2\beta$ implies

$$\lim_{k\to+\infty}\|\nabla f_2(x_k)-\nabla f_2(x^*)\|_2=0, \tag{3.21}$$

$$\lim_{k\to+\infty}\|B^T(v_k-v^*)\|_2=0, \tag{3.22}$$

$$\lim_{k\to+\infty}\|v_{k+1}-v_k\|_M=0. \tag{3.23}$$

By the definitions in (3.11) and (3.12),

$$\|v_{k+1}-v_k\|_2^2=\|v_{k+1}-v_k\|_M^2+\lambda\|B^T(v_{k+1}-v_k)\|_2^2,$$

which combined with (3.22) and (3.23) gives

$$\lim_{k\to+\infty}\|v_{k+1}-v_k\|_2=0. \tag{3.24}$$

On the other hand, from (3.8) we have

$$-\gamma\nabla f_2(x^*)-\lambda B^T v^*=0,$$

and from (2.2b)

$$x_{k+1}-x_k=-\gamma\nabla f_2(x_k)-\lambda B^T v_{k+1}.$$

Hence,

$$x_{k+1}-x_k=-\gamma(\nabla f_2(x_k)-\nabla f_2(x^*))-\lambda(B^T v_{k+1}-B^T v^*).$$

Now, using (3.21) and (3.22) we immediately get

$$\lim_{k\to+\infty}\|x_{k+1}-x_k\|_2=0. \tag{3.25}$$

By the definition in (3.13) and (3.24)-(3.25),

$$\lim_{k\to+\infty}\|u_{k+1}-u_k\|_\lambda=0. \tag{3.26}$$

From (3.20), we know that the sequence $\{\|u_k-u^*\|_\lambda\}$ is non-increasing, so the sequence $\{u_k\}$ is bounded and there exists a convergent subsequence $\{u_{k_j}\}$ such that

$$\lim_{j\to+\infty}\|u_{k_j}-\overline{u}^*\|_\lambda=0 \tag{3.27}$$

for some $\overline{u}^*\in\mathbb{R}^m\times\mathbb{R}^n$. Next, let us show that $\overline{u}^*$ is a fixed point of T. In fact,

$$\|T(u_{k_j})-\overline{u}^*\|_\lambda=\|(u_{k_j+1}-u_{k_j})-(u_{k_j}-\overline{u}^*)\|_\lambda\le\|u_{k_j+1}-u_{k_j}\|_\lambda+\|u_{k_j}-\overline{u}^*\|_\lambda, \tag{3.28}$$

which, in conjunction with (3.26) and (3.27), leads to

$$\lim_{j\to+\infty}\|T(u_{k_j})-\overline{u}^*\|_\lambda=0. \tag{3.29}$$

The operator T is continuous since it is nonexpansive, so it follows from (3.27) and (3.29) that $\overline{u}^*$ is a fixed point of T. Moreover, we have known that $\{\|u_k-u^*\|_\lambda\}$ is non-increasing for any fixed point u^* of T. In particular, by choosing $u^*=\overline{u}^*$, we see that $\{\|u_k-\overline{u}^*\|_\lambda\}$ is non-increasing. Combining this and (3.27) yields

$$\lim_{k\to+\infty}u_k=\overline{u}^*.$$

Writing $\overline{u}^* = (\overline{v}^*,\ \overline{x}^*)$ with $\overline{v}^* \in \mathbb{R}^m$, $\overline{x}^* \in \mathbb{R}^n$, we find from Theorem 3.1 that $\overline{x}^*$ is the solution of problem (1.1). ■

Note that if $f_2(x) = \frac{1}{2}\|x-b\|_2^2$ and $\gamma = 1$, then PDFP^{2}O reduces to FP^{2}O for $\kappa = 0$. As a consequence of the above theorem, we can achieve the convergence of FP^{2}O for $\kappa = 0$ even when BB^T is singular, for which no convergence is available from Theorem 3.12 of [21].

Corollary 3.6 *Suppose $0 < \lambda \le 1/\lambda_{\max}(BB^T)$. Let $\{v_k\}$ be the sequence generated by FP^{2}O for $\kappa = 0$. Set $x_k = b - \lambda B^T v_k$. Then the sequence $\{v_k\}$ converges to the fixed point of H (see (2.1)), the sequence $\{x_k\}$ converges to the solution of problem (1.1) with $f_2(x) = \frac{1}{2}\|x-b\|_2^2$.*

3.2 Linear convergence rate for special cases

In this section, we will give some stronger theoretical results about the convergence rate in some special cases. For this, we present the following condition.

Condition 3.1 *For any two real numbers λ and γ satisfying that $0 < \gamma < 2\beta$ and $0 < \lambda \le 1/\lambda_{\max}(BB^T)$, there exist $\eta_1,\ \eta_2 \in [0,1)$ such that $\|I - \lambda BB^T\|_2 \le \eta_1^2$ and*

$$\|g(x) - g(y)\|_2 \le \eta_2\|x-y\|_2 \quad \text{for all } x,\ y \in \mathbb{R}^n, \tag{3.30}$$

where $g(x)$ is given in (3.10).

Remark 3.2 *If B has full row rank, f_2 is strongly convex, i.e. there exists some $\sigma > 0$ such that*

$$\langle \nabla f_2(x) - \nabla f_2(y), x-y\rangle \ge \sigma\|x-y\|_2^2 \quad \text{for all } x,\ y \in \mathbb{R}^n, \tag{3.31}$$

then this condition can be satisfied. In fact, when B has full row rank, we can choose

$$\eta_1^2 = 1 - \lambda\lambda_{\min}(BB^T),$$

where $\lambda_{\min}(BB^T)$ denotes the smallest eigenvalue of BB^T. In this case, η_1^2 takes its minimum

$$(\eta_1^2)_{\min} = 1 - \frac{\lambda_{\min}(BB^T)}{\lambda_{\max}(BB^T)}$$

at $\lambda = 1/\lambda_{\max}(BB^T)$. On the other hand, since f_2 has $1/\beta$-Lipschitz continuous gradient and is strongly convex, it follows from (3.2) and (3.31) that

$$\begin{aligned}
&\|g(x) - g(y)\|_2^2 \\
=&\|x-y\|_2^2 - 2\gamma\langle \nabla f_2(x) - \nabla f_2(y), x-y\rangle + \gamma^2\|\nabla f_2(x) - \nabla f_2(y)\|_2^2 \\
\le&\|x-y\|_2^2 - \frac{\gamma(2\beta-\gamma)}{\beta}\langle \nabla f_2(x) - \nabla f_2(y), x-y\rangle \\
\le&(1 - \frac{\sigma\gamma(2\beta-\gamma)}{\beta})\|x-y\|_2^2.
\end{aligned} \tag{3.32}$$

Hence we can choose

$$\eta_2^2 = 1 - \frac{\sigma\gamma(2\beta-\gamma)}{\beta}.$$

In particular, if we choose $\gamma = \beta$, then η_2 takes its minimum in the present form:

$$(\eta_2^2)_{\min} = 1 - \sigma\beta.$$

As a typical example, consider $f_2(x) = \frac{1}{2}\|Ax-b\|_2^2$ with A^TA full rank. Then we can find out that $\beta = 1/\lambda_{\max}(A^TA)$ and $\sigma = \lambda_{\min}(A^TA)$, and hence

$$(\eta_2^2)_{\min} = 1 - \frac{\lambda_{\min}(A^TA)}{\lambda_{\max}(A^TA)}.$$

Despite most of our interesting problems do not belong to these special cases, and there will be more efficient algorithms if the condition 3.1 is satisfied, the following results still have some theoretical values where the best performance of PDFP^{2}O$_\kappa$ can be achieved. First of all, we show that T_κ is contractive under condition 3.1.

Lemma 3.5 *Suppose condition 3.1 holds true. Let the operator T be given in* (3.5) *and $T_\kappa = \kappa I + (1-\kappa)T$ for $\kappa \in [0,1)$. Then T_κ is contractive under the norm $\|\cdot\|_\lambda$.*

Proof. Let $\eta = \max\{\eta_1, \eta_2\}$. It is clear that $0 < \eta < 1$. Then, owing to the condition 3.1, for all $u_1 = (v_1, x_1)$, $u_2 = (v_2, x_2) \in \mathbb{R}^m \times \mathbb{R}^n$, there holds

$$\begin{aligned} \|g(x_1) - g(x_2)\|_2 \leq & \eta \|x_1 - x_2\|_2, \\ \|v_1 - v_2\|_M \leq & \eta \|v_1 - v_2\|_2, \end{aligned}$$

from which, (3.13) and (3.17) it follows that

$$\begin{aligned} & \|T(u_1) - T(u_2)\|_\lambda^2 \\ \leq & \|g(x_1) - g(x_2)\|_2^2 + \lambda \|v_1 - v_2\|_M^2 - \lambda \|(T_1(u_1) - T_1(u_2)) - (v_1 - v_2)\|_M^2 \\ \leq & \eta^2 (\|x_1 - x_2\|_2^2 + \lambda \|v_1 - v_2\|_2^2) \\ = & \eta^2 \|u_1 - u_2\|_\lambda^2. \end{aligned}$$

On the other hand, it is easy to check from the last estimate and the triangle inequality that

$$\|T_\kappa(u_1) - T_\kappa(u_2)\|_\lambda \leq \kappa \|u_1 - u_2\|_\lambda + (1-\kappa)\|T(u_1) - T(u_2)\|_\lambda \leq \theta \|u_1 - u_2\|_\lambda,$$

with $\theta = \kappa + (1-\kappa)\eta \in (0,1)$. So operator T_κ is contractive. ■

Now, we are ready to analyze the convergence rate of PDFP^{2}O$_\kappa$.

Theorem 3.7 *Suppose condition 3.1 holds true. Let the operator T be given in* (3.5) *and $T_\kappa = \kappa I + (1-\kappa)T$ for $\kappa \in [0,1)$. Let $u_k = (v_k, x_k)$ be a Picard sequence of operator T_κ (or equivalently, a sequence obtained by algorithm PDFP^{2}O$_\kappa$). Then the sequence $\{u_k\}$ must converge to the unique fixed point $u^* = (v^*,\ x^*) \in \mathbb{R}^m \times \mathbb{R}^n$ of T, with x^* the unique solution of problem* (1.1). *Furthermore, there holds the estimate*

$$\|x_k - x^*\|_2 \leq \frac{c\theta^k}{1-\theta}, \tag{3.33}$$

where $c = \|u_1 - u_0\|_\lambda$, $\theta = \kappa + (1-\kappa)\eta \in (0,1)$ and $\eta = \max\{\eta_1, \eta_2\}$, with η_1 and η_2 given in condition 3.1.

Proof. Since the operator T_κ is contractive, by the Banach contraction mapping theorem, it has a unique fixed point, denoted by $u^* = (v^*,\ x^*)$. It is obvious that T_κ has same fixed points as T, so x^* is the unique solution of problem (1.1) from Theorem 3.1. Moreover, it is routine that the sequence $\{u_k\}$ converges to u^*. On the other hand, it follows from Lemma 3.5 that

$$\|u_{k+1} - u_k\|_\lambda \leq \theta \|u_k - u_{k-1}\|_\lambda \leq \cdots \leq \theta^k \|u_1 - u_0\|_\lambda = c\theta^k.$$

So for all $0 < l \in \mathbb{N}$,

$$\|u_{k+l} - u_k\|_\lambda \leq \sum_{i=1}^{l} \|u_{k+i} - u_{k+i-1}\|_\lambda = c\theta^k \sum_{i=1}^{l} \theta^{i-1} < \frac{c\theta^k}{1-\theta},$$

which immediately implies

$$\|x_k - x^*\|_2 \leq \|u_k - u^*\|_\lambda \leq \frac{c\theta^k}{1-\theta}$$

by letting $l \to +\infty$. The desired estimate (3.33) is then obtained. ■

If $B = I$, $\lambda = 1$, then form (2.2) is equivalent to form (1.4), so as a special case of Theorem 3.7, we can get the convergence rate for PFBS.

Corollary 3.8 *Suppose $0 < \gamma < 2\beta$ and there exists $\eta \in (0,1)$ such that*

$$\|g(x) - g(y)\|_2 \leq \eta\|x - y\|_2 \quad \text{for all } x,\ y \in \mathbb{R}^n.$$

Let $\{x_k\}$ be a sequence generated by PFBS and x^* the solution of problem (1.3) for $\mathcal{X} = \mathbb{R}^n$. Set $c = \|x_1 - x_0\|_2$. Then

$$\|x_k - x^*\|_2 \leq \frac{c\eta^k}{1-\eta}.$$

As a conclusion of Theorem 3.7, we can also get the convergence rate of FP^{2}O for $\kappa = 0$ under the assumption $\|I - \lambda BB^T\| < 1$.

Corollary 3.9 *Suppose $0 < \lambda \leq 1/\lambda_{\max}(BB^T)$, the matrix B has full row rank, and η_1 is given in condition 3.1. Let v^* be the fixed point of H (cf. (2.1)). Let $\{v_k\}$ be a sequence generated by FP^{2}O for $\kappa = 0$, with $x_k = b - \lambda B^T v_k$. Set*

$$c = \sqrt{\|\lambda B^T(v_1 - v_0)\|_2^2 + \lambda\|v_1 - v_0\|_2^2}.$$

Then

$$\|v_k - v^*\|_2 \leq \frac{c\eta_1^k}{\sqrt{\lambda}(1-\eta_1)}, \quad \|x_k - x^*\|_2 \leq \frac{c\eta_1^k}{1-\eta_1}.$$

4 Connections to other algorithms

We will further investigate the proposed algorithm PDFP^{2}O from the perspective of primal-dual forms and establish the connections to other existing methods.

4.1 Primal-dual and proximal point algorithms

For the problem (1.1), we can write its primal-dual form using Fenchel duality [29] as

$$\min_x \max_{\overline{v}} L(x, \overline{v}) := f_2(x) + \langle Bx, \overline{v}\rangle - f_1^*(\overline{v}), \tag{4.1}$$

where f_1^* is the convex conjugate function of f_1 defined by

$$f_1^*(\overline{v}) = \sup_{\overline{w} \in \mathbb{R}^m} \langle \overline{v}, \overline{w}\rangle - f_1(\overline{w}).$$

By introducing new intermediate variable y_{k+1}, equation (2.2) are reformulated as

$$\begin{cases} y_{k+1} = x_k - \gamma\nabla f_2(x_k) - \lambda B^T v_k, & \text{(4.2a)} \\ v_{k+1} = (I - \operatorname{prox}_{\frac{\gamma}{\lambda}f_1})(By_{k+1} + v_k), & \text{(4.2b)} \\ x_{k+1} = x_k - \gamma\nabla f_2(x_k) - \lambda B^T v_{k+1}. & \text{(4.2c)} \end{cases}$$

According to Moreau decomposition (see equation (2.21) in [13]), for all $v \in \mathbb{R}^m$, we have

$$v = v_{\frac{\gamma}{\lambda}}^{\oplus} + v_{\frac{\gamma}{\lambda}}^{\ominus}, \text{ where } v_{\frac{\gamma}{\lambda}}^{\oplus} = \operatorname{prox}_{\frac{\gamma}{\lambda}f_1} v,\ v_{\frac{\gamma}{\lambda}}^{\ominus} = \frac{\gamma}{\lambda}\operatorname{prox}_{\frac{\lambda}{\gamma}f_1^*}(\frac{\lambda}{\gamma}v),$$

from which we know

$$(I - \text{prox}_{\frac{\gamma}{\lambda} f_1})(By_{k+1} + v_k) = \frac{\gamma}{\lambda}\text{prox}_{\frac{\lambda}{\gamma} f_1^*}(\frac{\lambda}{\gamma}By_{k+1} + \frac{\lambda}{\gamma}v_k).$$

Let $\overline{v}_k = \frac{\lambda}{\gamma}v_k$. Then (4.2) can be reformulated as

$$\begin{cases} y_{k+1} = x_k - \gamma\nabla f_2(x_k) - \gamma B^T\overline{v}_k, & (4.3a) \\ \overline{v}_{k+1} = \text{prox}_{\frac{\lambda}{\gamma} f_1^*}(\frac{\lambda}{\gamma}By_{k+1} + \overline{v}_k), & (4.3b) \\ x_{k+1} = x_k - \gamma\nabla f_2(x_k) - \gamma B^T\overline{v}_{k+1}. & (4.3c) \end{cases}$$

In terms of the saddle point formulation (4.1), we have by a direct manipulation that

$$\begin{aligned} \nabla f_2(x_k) + B^T\overline{v}_k &= \nabla_x L(x_k, \overline{v}_k), \\ \text{prox}_{\frac{\lambda}{\gamma} f_1^*}(\frac{\lambda}{\gamma}By_{k+1} + \overline{v}_k) &= \arg\min_{\overline{v}\in\mathbb{R}^m} - L(y_{k+1}, \overline{v}) + \frac{\gamma}{2\lambda}\|\overline{v} - \overline{v}_k\|_2^2, \\ \nabla f_2(x_k) + B^T\overline{v}_{k+1} &= \nabla_x L(x_k, \overline{v}_{k+1}). \end{aligned}$$

Hence, (4.3) can be expressed as

$$\begin{cases} y_{k+1} = x_k - \gamma\nabla_x L(x_k, \overline{v}_k), & (4.4a) \\ \overline{v}_{k+1} = \arg\min_{\overline{v}\in\mathbb{R}^m} - L(y_{k+1}, \overline{v}) + \frac{\gamma}{2\lambda}\|\overline{v} - \overline{v}_k\|_2^2, & (4.4b) \\ x_{k+1} = x_k - \gamma\nabla_x L(x_k, \overline{v}_{k+1}). & (4.4c) \end{cases}$$

From (4.3a) and (4.3c), we can find out that

$$y_{k+1} = x_{k+1} + \gamma B^T(\overline{v}_{k+1} - \overline{v}_k).$$

Then the equation (4.4b) becomes

$$\overline{v}_{k+1} = \arg\min_{\overline{v}\in\mathbb{R}^m} - L(x_{k+1}, \overline{v}) + \frac{\gamma}{2\lambda}\|\overline{v} - \overline{v}_k\|_M^2,$$

where $M = 1 - \lambda BB^T$. Together with the (4.4c), the iterations (4.4) are

$$\begin{cases} \overline{v}_{k+1} = \arg\max_{\overline{v}\in\mathbb{R}^m} L(x_{k+1}, \overline{v}) - \frac{\gamma}{2\lambda}\|\overline{v} - \overline{v}_k\|_M^2, \\ x_{k+1} = x_k - \gamma\nabla_x L(x_k, \overline{v}_{k+1}). \end{cases}$$

Thus the proposed algorithm can be interpreted as an inexact Uzawa method [2] applied on the dual formulation. Compared to classical Uzawa method, this proposed method is more implicit since the update of $\overline{v}_{k+1}$ involves x_{k+1} and a proximal point iteration matrix M is used.

This leads to a close connection to a class of primal-dual method studied in [37, 16, 9, 19]. For example, in [9], Chambolle and Pock proposed the following scheme for solving (4.1):

$$\text{(CP)} \quad \begin{cases} \overline{v}_{k+1} = (I + \sigma\partial f_1^*)^{-1}(\overline{v}_k + \sigma By_k), & (4.6a) \\ x_{k+1} = (I + \tau\nabla f_2)^{-1}(x_k - \tau B^T\overline{v}_{k+1}), & (4.6b) \\ y_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k), & (4.6c) \end{cases}$$

where $\sigma, \tau > 0$, $\theta \in [0, 1]$ is a parameter. For $\theta = 0$, we can get the classical Arrow-Hurwicz-Uzawa (AHU) method in [2]. The convergence of AHU with very small step length is shown in [16]. Under some assumptions on f_1^* or strongly convexity of f_2, global convergence of the primal-dual gap can also be shown with specific chosen adaptive steplength [9]. Note that in the case of ROF model, Chan and

Zhu proposed in [38] a clever adaptive step lengths σ and τ for acceleration, and recently the convergence is shown in [4].

According to equation (4.3), using the relation $\text{prox}_{\frac{\lambda}{\gamma}f_1^*} = (I + \frac{\lambda}{\gamma}\partial f_1^*)^{-1}$, and changing the order of these equations, we know that PDFP^{2}O is equivalent to

$$\begin{cases} \overline{v}_{k+1} = (I + \dfrac{\lambda}{\gamma}\partial f_1^*)^{-1}(\dfrac{\lambda}{\gamma}By_k + \overline{v}_k), & \text{(4.7a)} \\ x_{k+1} = x_k - \gamma\nabla f_2(x_k) - \gamma B^T\overline{v}_{k+1}, & \text{(4.7b)} \\ y_{k+1} = x_{k+1} - \gamma\nabla f_2(x_{k+1}) - \gamma B^T\overline{v}_{k+1}. & \text{(4.7c)} \end{cases}$$

Let $\sigma = \lambda/\gamma$, $\tau = \gamma$, we can see that equations (4.6b) and (4.6c) are approximated by two explicit steps (4.7b)-(4.7c). In summary, we list the comparisons of CP for $\theta = 1$ with fixed step length and PDFP^{2}O in Table 4.1.

Table 4.1: The comparison between CP ($\theta = 1$) and PDFP^{2}O.

	CP ($\theta = 1$)	PDFP^{2}O
Form	$\overline{v}_{k+1} = (I + \sigma\partial f_1^*)^{-1}(\overline{v}_k + \sigma By_k)$ $x_{k+1} = (I + \tau\nabla f_2)^{-1}(x_k - \tau B^T\overline{v}_{k+1})$ $y_{k+1} = 2x_{k+1} - x_k$	$\overline{v}_{k+1} = (I + \frac{\lambda}{\gamma}\partial f_1^*)^{-1}(\frac{\lambda}{\gamma}\overline{v}_k + By_k)$ $x_{k+1} = x_k - \gamma\nabla f_2(x_k) - \gamma B^T\overline{v}_{k+1}$ $y_{k+1} = x_{k+1} - \gamma\nabla f_2(x_{k+1}) - \gamma B^T\overline{v}_{k+1}$
Convergence	$0 < \sigma\tau < 1/\lambda_{\max}(BB^T)$	$0 < \gamma < 2\beta$, $0 < \lambda \leq 1/\lambda_{\max}(BB^T)$
Relation	$\sigma = \lambda/\gamma$, $\tau = \gamma$	

For $f_2(x) = \frac{1}{2}\|Ax - b\|_2^2$, (4.4) can be further expressed as

$$\begin{cases} y_{k+1} = \arg\min\limits_{x\in\mathbb{R}^n} L(x, \overline{v}_k) + \dfrac{1}{2\gamma}\|x - x_k\|^2_{(I-\gamma A^TA)}, & \text{(4.8a)} \\ \overline{v}_{k+1} = \arg\min\limits_{\overline{v}\in\mathbb{R}^m} - L(y_{k+1}, \overline{v}) + \dfrac{\gamma}{2\lambda}\|\overline{v} - \overline{v}_k\|_2^2, & \text{(4.8b)} \\ x_{k+1} = \arg\min\limits_{x\in\mathbb{R}^n} L(x, \overline{v}_{k+1}) + \dfrac{1}{2\gamma}\|x - x_k\|^2_{(I-\gamma A^TA)}. & \text{(4.8c)} \end{cases}$$

Note that by introducing the proximal iteration norm through the matrix $I - \gamma A^TA \in \mathrm{M}_n$ for $0 < \gamma < \beta$ with $\beta = \lambda_{\max}(A^TA)$, (4.8a) and (4.8c) become explicit. This is particularly useful when the inverse of A^TA is not easy to obtain in most of imaging applications, such as superresolution, tomographic reconstruction and parallel MRI [12]. Meanwhile, it worths to point out that the condition on γ by this formulation is stricter than Theorem 3.5, where γ is required as $0 < \gamma < 2\beta$ for the convergence. Furthermore, if we denote $\hat{u}_k = \left(\overline{v}_k{}^T, x_k{}^T\right)^T$ and $F(\hat{u}_k) = \begin{pmatrix} \partial f_1^*(\overline{v}_k) - Bx_k \\ B^T\overline{v}_k + \nabla f_2(x_k) \end{pmatrix}$ and $P = \begin{pmatrix} \frac{\gamma}{\lambda}(I - \lambda BB^T) & 0 \\ 0 & \frac{1}{\gamma}(I - \gamma A^TA) \end{pmatrix}$, we can also easily write the algorithm in the proximal point algorithm (PPA) framework [19] as

$$0 \in F(\hat{u}_{k+1}) + P(\hat{u}_{k+1} - \hat{u}_k). \tag{4.9}$$

We note that in [19], Chambolle-Pock algorithm (4.6) for $\theta = 1$ was also rewritten in PPA structure as (4.9) with the same F while

$$P = \begin{pmatrix} \frac{1}{\sigma}I & B \\ B^T & \frac{1}{\tau}I \end{pmatrix}.$$

In [19, 9], more general class of algorithms taking this form are studied. In particular, extra extrapolation step can be applied to the algorithm (4.9) for acceleration.

4.2 Splitting type of methods

There are other types of methods which are designed to solve the problem (1.1) based on the notion of augmented Lagrangian. For simplicity, we only list these algorithms for $f_2(x) = \frac{1}{2}\|Ax - b\|^2$. Among them, the alternating split Bregman (ASB) method proposed by Goldstein and Osher [18] is very popular for imaging applications. This method has been proved to be equivalent to Douglas-Rachford method and alternating direction of multiplier method (ADMM). In [36, 37], based on PFBS and Bregman iteration, a split inexact Uzawa (SIU) method is proposed to maximally decouple the iterations so that each iteration is explicit. Further analysis and connections to primal-dual methods algorithm are given in [16, 37]. In particular, it is shown that the primal-dual algorithm scheme (4.6) with $\theta = 1$ can be interpreted as SIU.

In the following, we study the connections and differences to these two methods. ASB can be described as follows:

$$\text{(ASB)} \quad \begin{cases} x_{k+1} = (A^TA + \nu B^TB)^{-1}(A^Tb + \nu B^T(d_k - v_k)), & \text{(4.10a)} \\ d_{k+1} = \text{prox}_{\frac{1}{\nu}f_1}(Bx_{k+1} + v_k), & \text{(4.10b)} \\ v_{k+1} = v_k - (d_{k+1} - Bx_{k+1}), & \text{(4.10c)} \end{cases}$$

where $\nu > 0$ is a parameter. The explicit SIU method proposed in the literature [37] can be described as

$$\text{(SIU)} \quad \begin{cases} x_{k+1} = x_k - \delta A^T(Ax_k - b) - \delta\nu B^T(Bx_k - d_k + v_k), & \text{(4.11a)} \\ d_{k+1} = \text{prox}_{\frac{1}{\nu}f_1}(Bx_{k+1} + v_k), & \text{(4.11b)} \\ v_{k+1} = v_k - (d_{k+1} - Bx_{k+1}), & \text{(4.11c)} \end{cases}$$

where $\delta > 0$ is a parameter. We can easily see that we approximate the implicit step (4.10a) in ASB by an explicit step (4.11a) in SIU.

From (4.2a) and (4.2c), we can find out a relation of y_k and x_k, given by

$$x_k = y_k - \lambda B^T(v_k - v_{k-1}).$$

Then eliminating x_k, PDFP^{2}O can be expressed as

$$\begin{cases} y_{k+1} = y_k - \lambda B^T(2v_k - v_{k-1}) - \gamma\nabla f_2(y_k - \lambda B^T(v_k - v_{k-1})), & \text{(4.12a)} \\ v_{k+1} = (I - \text{prox}_{\frac{\gamma}{\lambda}f_1})(By_{k+1} + v_k). & \text{(4.12b)} \end{cases}$$

By inserting the splitting variable d_{k+1} in (4.12b), (4.12) can be further expressed as

$$\begin{cases} y_{k+1} = y_k - \lambda B^T(By_k - d_k + v_k) - \gamma\nabla f_2(y_k - \lambda B^T(By_k - d_k)), \\ d_{k+1} = \text{prox}_{\frac{\gamma}{\lambda}f_1}(By_{k+1} + v_k), \\ v_{k+1} = v_k - (d_{k+1} - By_{k+1}). \end{cases} \tag{4.13}$$

For $f_2(x) = \frac{1}{2}\|Ax - b\|_2^2$, $\nabla f_2(x) = A^T(Ax - b)$. By changing the order and letting $\gamma = \delta$, $\lambda = \delta\nu$, (4.13) becomes

$$\begin{cases} y_{k+1} = y_k - \delta A^T(Ay_k - b) - \delta\nu B^T(By_k - d_k + v_k) - \delta^2\nu A^TAB^T(d_k - By_k) & \text{(4.14a)} \\ d_{k+1} = (\text{prox}_{\frac{1}{\nu}f_1})(By_{k+1} + v_k), & \text{(4.14b)} \\ v_{k+1} = v_k - (d_{k+1} - By_{k+1}). & \text{(4.14c)} \end{cases}$$

We can easily see that equation (4.10a) in ASB is approximated by (4.14a). Although it seems that PDFP^{2}O requires more computation in (4.14a) than SIU in (4.11a), PDFP^{2}O has the same computation

cost as that of SIU if the iterations are implemented cleverly. For the reason of comparison, we can change the variable y_k to x_k in (4.14). Table 4.2 gives the summarized comparisons among ASB, SIU and PDFP^{2}O. We can see that the only difference of SIU and PDFP2O is on the first step. As two algorithms converges, the algorithm PDFP2O behaves asymptotically the same as SIU since $d_k - Bx_k$ converges to 0. The parameters δ and ν satisfy respectively different conditions to ensure the convergence.

Table 4.2: The comparisons among ASB, SIU and PDFP^{2}O.

	ASB	SIU	PDFP^{2}O
Form	$x_{k+1} = (A^TA + \nu B^TB)^{-1}$ $(A^Tb + \nu B^T(d_k - v_k))$ $d_{k+1} = \text{prox}_{\frac{1}{\nu}f_1}(Bx_{k+1} + v_k)$ $v_{k+1} = v_k - (d_{k+1} - Bx_{k+1})$	$x_{k+1} = x_k - \delta A^T(Ax_k - b)$ $-\delta\nu B^T(Bx_k - d_k + v_k)$ $d_{k+1} = \text{prox}_{\frac{1}{\nu}f_1}(Bx_{k+1} + v_k)$ $v_{k+1} = v_k - (d_{k+1} - Bx_{k+1})$	$x_{k+1} = x_k - \delta A^T(Ax_k - b)$ $-\delta\nu B^T(Bx_k - d_k + v_k)$ $-\delta^2\nu A^TAB^T(d_k - Bx_k)$ $d_{k+1} = \text{prox}_{\frac{1}{\nu}f_1}(Bx_{k+1} + v_k)$ $v_{k+1} = v_k - (d_{k+1} - Bx_{k+1})$
Conver -gence	$\nu > 0$	$\nu > 0$ $0 < \delta \le 1/\lambda_{\max}(A^TA + \nu B^TB)$	$0 < \delta < 2/\lambda_{\max}(A^TA)$ $0 < \delta\nu \le 1/\lambda_{\max}(BB^T)$

5 Numerical experiments

In this section, we will show the efficiency of PDFP^{2}O$_\kappa$ for $\kappa \in [0, 1)$ through two applications: image superresolution and computerized tomography (CT) reconstruction. These two applications can be both described as problem (1.2), where A is a linear operator representing subsampling and tomographic projection operator respectively. Here, we use total variation as the regularization functional, where the operator $B : R^n \to R^{2n}$ is a discrete gradient operator. Furthermore, the isotropic definition is adopted, i.e. $f_1(w) = \mu\|w\|_{1,2}$, for all $w = (w_1, \ldots, w_n, w_{n+1}, \cdots, w_{2n})^T \in \mathbb{R}^{2n}$,

$$\|w\|_{1,2} = \sum_{i=1}^{n} \sqrt{w_i^2 + w_{n+i}^2}.$$

Let $\mathbf{w}_i = (w_i, w_{n+i})^T$, $\|\mathbf{w}_i\|_2 = \sqrt{w_i^2 + w_{n+i}^2}$ and $\varepsilon = \frac{\mu\gamma}{\lambda}$. Then $\text{prox}_{\varepsilon\|\cdot\|_{1,2}}(w)$ can be expressed as

$$(\text{prox}_{\varepsilon\|\cdot\|_{1,2}}(w))_{i,n+i} = \max\{\|\mathbf{w}_i\|_2 - \varepsilon, 0\}\frac{\mathbf{w}_i}{\|\mathbf{w}_i\|_2},\ i = 1, \cdots, n.$$

For the implementation of PDFP2O, we use the scheme presented in Algorithm 4, where we compute directly $(I - \text{prox}_{\frac{\gamma}{\lambda}f_1})(w)$. In fact, we can deduce that $(I - \text{prox}_{\frac{\gamma}{\lambda}f_1})(w) = \text{Proj}_\varepsilon(w)$, where Proj_ε is the projection operator from $\mathbb{R}^{2n}$ to $\ell_{2,\infty}$ ball of radius ε, i.e.

$$(\text{Proj}_\varepsilon(w))_{i,n+i} = \min\{\|\mathbf{w}_i\|_2, \varepsilon\}\frac{\mathbf{w}_i}{\|\mathbf{w}_i\|_2},\ i = 1, \cdots, n.$$

5.1 Image superresolution

In the numerical simulation, the subsampling operator A is implemented by taking the average of every $d \times d$ pixels and sampling the average, if a zoom in ratio d is desired. The experiment is performed on the test image "lena" of size 512×512 and the subsampling ratio is $d = 4$. White Gaussian noise of mean 0 and variance 1 is added to the observed low resolution image of 128×128. The regularization parameter μ is set as 0.1 for the best image quality.

First we show the impacts of the parameters κ, γ and λ for the proposed algorithm in Figure 5.1. The conditions for theoretical convergence are $0 < \gamma < 2\beta$, $0 < \lambda \le 1/\lambda_{\max}(BB^T)$ and $\kappa \in [0, 1)$ (see

Theorems 3.4 and 3.5). We can see that the maximum eigenvalue of A^TA is 1/16, thus by the discussion in Remark 3.2, we have $0 < \gamma < 32$. It is well known in total variation application that $\lambda_{\max}(BB^T) = 8$ for B being the usual gradient operator (see [16]), then $0 < \lambda \leq 1/8$. From Figure 5.1 (a) and (b), we can see that for most cases $\kappa = 0$ achieves the fastest convergence compared to other $\kappa \in (0, 1)$, so we choose $\kappa = 0$ for the following comparison. In Figure 5.1 (c) and (d), the parameter λ has relatively smaller impact on the performance of this algorithm. We compare the results for $\lambda = 1/5, 1/6, 1/8, 1/16, 1/32$. When $\lambda = 1/6 > 1/8$, the algorithm is convergent. While for $\lambda = 1/5$, the algorithm does not appear to converge, which shows that we can not extend the range of λ to $(0, 2/\lambda_{\max}(BB^T))$ generally. Hence, we only consider $0 < \lambda \leq 1/\lambda_{\max}(BB^T)$ as indicated in Theorem 3.5, for which the upper bound $\lambda = 1/8$ achieves the best performance. The parameter γ has larger impact for the algorithm. We test $\gamma = 8, 16, 24, 30, 32$ for $\kappa = 0$, $\lambda = 1/8$. Numerically, larger γ leads to a faster convergence, but when γ is very close to $2\beta = 32$, the energy starts to oscillate (Figure 5.1 (e) and (f)). For this reason, we choose $\gamma = 30$.

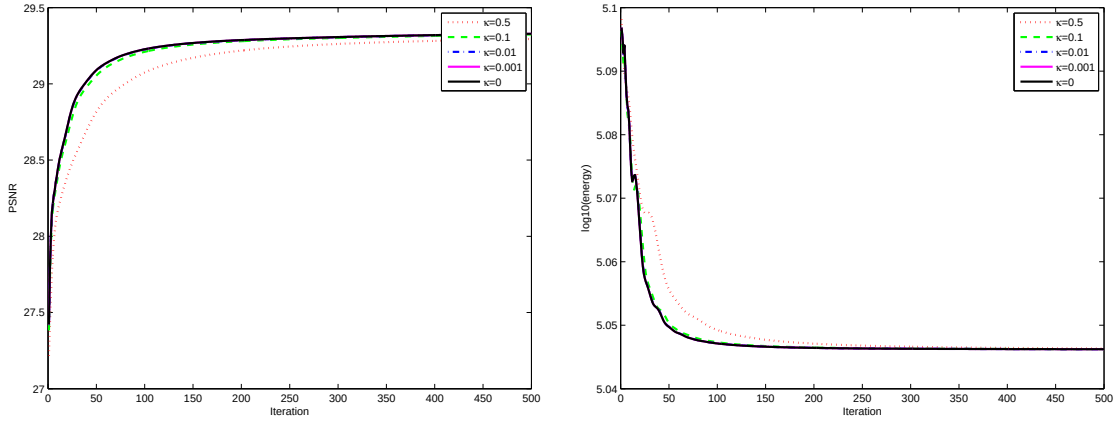
In the following, we compare the performance of ASB, SIU and PDFP2O. Both PDFP2O and SIU only require matrix-vector multiplications (see Table 4.2). For the implementation of ASB, we solve the inverse of the matrix $A^TA + \nu B^TB$ directly since fast Fourier transform can not be applied in this case. We use the image obtained by nearest neighbor interpolation as the initial guess. The parameter ν is crucial for the performance of ASB, and we have tested a range of ν and set $\nu = 0.01$ for a good balance of computational speed and image quality. For SIU, we set $\delta = 30$ and $\nu = 0.001$. In PDFP^{2}O, we choose $\gamma = 30$, $\lambda = 1/8$ as we discussed previously. The results are shown in Figure 5.2 and the images obtained are comparable. As we point out previously, the direct inverse of the linear equation (4.10a) in ASB might not be easy to obtain, while both PDFP2O and SIU have simple and explicit iterates. In Figure 5.3 (a), the comparison of energy decreasing *versus* iterations number and computation time are present in Figure 5.3 (a) and (b) respectively. As expected, one iteration for ASB requires more computation time than PDFP^{2}O and the energy by ASB decreases faster in terms of iterations number, while slower in terms of computation time. For the two explicit methods, PDFP2O is faster than SIU for both criteria.

5.2 CT reconstruction

In a simplified parallel tomographic problem, an observed body slice is modeled as a two dimensional function, and projections modeled by line integrals represents the total attenuation of a beam of x-rays when it traverses the object. The operator for this application can be represented by a discrete Radon transform, and the tomographic reconstruction problem is then to estimate a function from a finite number of measured line integrals(see [3]). The standard reconstruction algorithm in clinical applications is so-called Filtered Back Projection (FBP) algorithm. In the presence of noise, this problem becomes difficult since the inverse of Radon transform is unbounded and ill-posed. In the literature, the model (1.2) is often used for iterative reconstruction. Here A is the Radon transform matrix, b is the measured projections vector. Generally the size of A is huge and it is not easy to either compute the inverse directly or estimate the eigenvalues.

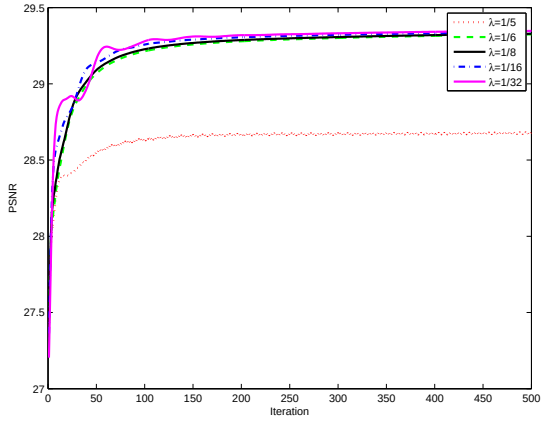
Here, we use the same example tested in [37], i.e., 50 uniformly oriented projections are simulated for a 128×128 Shepp-Logan phantom image and then white Gaussian noise of $\sigma = 1$ is added to the data. As the previous example, we first test the impacts of the parameters κ, γ and λ. The impact of κ has the same behavior as for superresolution example, i.e. $\kappa = 0$ is the best for $\kappa \in [0, 1)$ (see Figure 5.4 (a) and (b)). Similarly, the parameter λ has little impact on the performance of the algorithm (see Figure 5.4 (c) and (d)). On one hand, the convergence is guaranteed for $0 < \lambda \leq 1/8$, on the other

Figure 5.1: PSNR and energy *versus* iterations with different parameters. (a) and (b) are PSNR and energy *versus* iterations respectively for $\kappa = 0.5, 0.1, 0.01, 0.001, 0 (\lambda = 1/8,\ \gamma = 30)$. (c) and (d) are PSNR and energy *versus* iterations for $\lambda = 1/5, 1/6, 1/8, 1/16, 1/32$ ($\kappa = 0,\ \gamma = 30$). (e) and (f) are PSNR and energy *versus* iterations for $\gamma = 8, 16, 24, 30, 32$ ($\kappa = 0,\ \lambda = 1/8$).

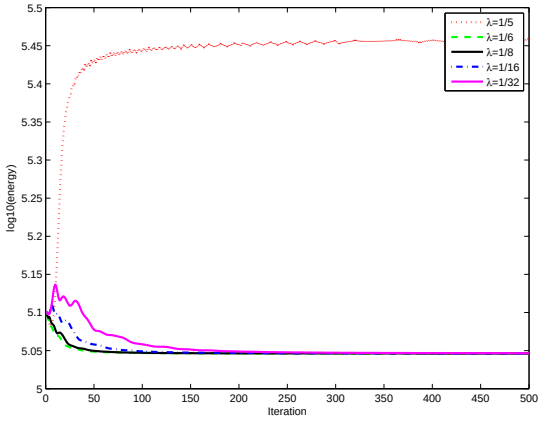


(a)

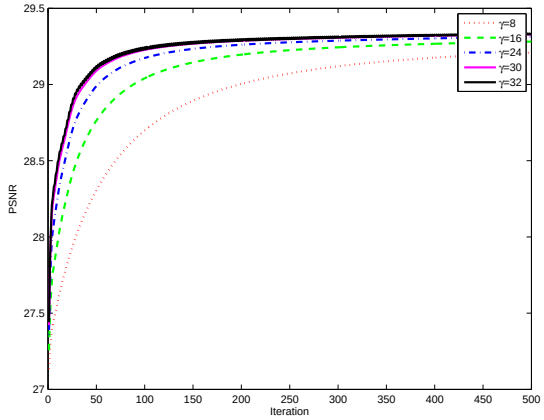
(b)



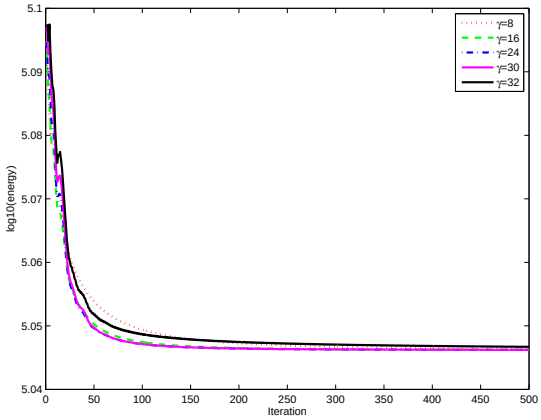
(c)



(d)



(e)



(f)

Figure 5.2: Super resolution results from 128×128 image to 512×512 image by ASB, SIU and PDFP^{2}O; noise level $\sigma = 1$.

Original

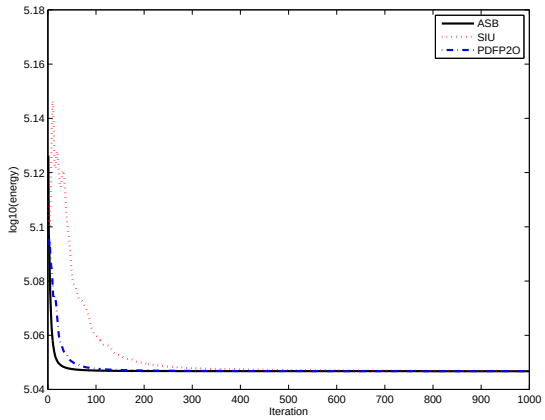
ASB, PSNR=29.3519

SIU, PSNR=29.3497

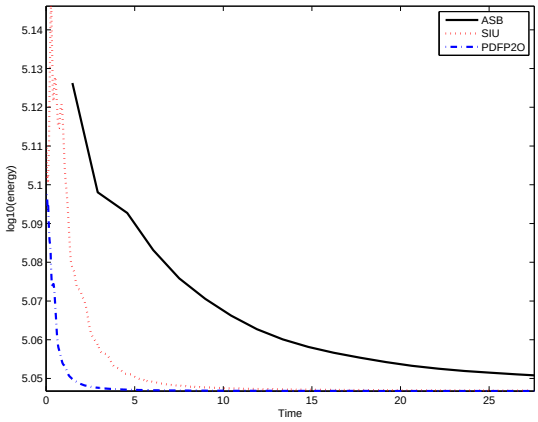
PDFP^{2}O, PSNR=29.3392

Figure 5.3: Comparisons of energy with respect to iterations (a) and time (b) for ASB, SIU and PDFP^{2}O.

(a) Energy *versus* Iterations



(b) Energy *versus* Time



hand the convergence still holds numerically for larger λ. After zooming out the iterations, we find out that energy decreases the fastest for $\lambda = 1/8$, while PSNR increases the fastest for $\lambda = 1/32$. Finally, the parameter γ has large impact on the convergence of the algorithm. Theoretically, it should satisfy $0 < \gamma < 2\beta$ where $1/\beta$ being the largest eigenvalue of $A^T A$. Numerically, we test $\gamma = 0.4, 0.7, 1, 1.2, 1.3$ for $\kappa = 0$, $\lambda = 1/8$. Better performance with a larger γ is observed (see Figure 5.4 (e) and (f)) while when $\gamma = 1.4$, the algorithm diverges.

In this example, due to the large size of the discrete Radon transform matrix, the inverse of $(A^T A + \nu B^T B)$ is not easy to compute, thus we do not draw ASB for comparison. Instead, we compare the performance of PDFP2O and SIU in Figure 5.5. The parameters are set as $\delta = 1.3$, $\nu = 0.1$ for SIU and $\gamma = 1.3$, $\lambda = 1/8$ for PDFP2O. The final reconstruction results are similar. Surprisedly, although the iterative schemes are different for the two algorithms, the evolution of PSNR and energy are very close for the two algorithms (see Figure 5.6) on this example.

6 Conclusions

In this paper, we design an efficient algorithm, called PDFP^{2}O$_\kappa$, for solving problem (1.1). We express it in a fixed point form to analyze its convergence for $\kappa \in [0, 1)$ in general cases. For some special cases, we further analyze the convergence rate of PDFP^{2}O$_\kappa$. To highlight the nature of PDFP^{2}O, we present some equivalent forms of PDFP^{2}O and reveal its connection and difference with other algorithms. For the implementation of PDFP^{2}O$_\kappa$, no linear systems are required to solve at each iteration, and the strategies for choosing involved parameters are also very clear. Hence, the computational cost for PDFP^{2}O$_\kappa$ is cheap and can be easily implemented in parallel, which is particulary useful for large scale inverse problems. Finally, we illustrate the efficiency of PDFP^{2}O$_\kappa$ through some numerical examples on image supperresolution and CT reconstruction. In conclusion, the present framework has simple iteration scheme and can be easily adapted to those inverse problems involving large scale and complicated functionals minimization.

Acknowledgments

JH was partially supported by the NNSFC (Grant nos. 11171219, 11161130004) and E–Institutes of Shanghai Municipal Education Commission (E03004). XZ was partially supported by the NNSFC (Grant nos. 11101277, 11161130004) and the Shanghai Pujiang Talent Program (Grant no. 11PJ1405900).

References

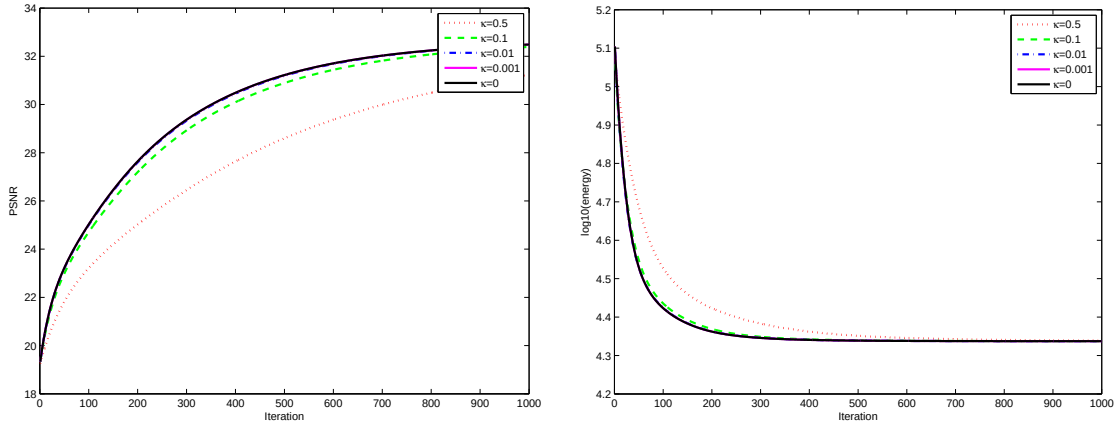
[1] Argyriou A, Micchelli C A, Pontil M, Shen L and Xu Y 2011 Efficient first order methods for linear composite regularizers *Arxiv preprint arXiv*:1104.1436

[2] Arrow K J, Hurwicz L and Uzawa H 1958 *Studies in Linear and Non-linear Programming* (Stanford: Stanford Univeristy Press)

[3] Avinash C and Malcolm S 2001 *Principles of Computerized Tomographic Imaging* (Philadelphia: SIAM)

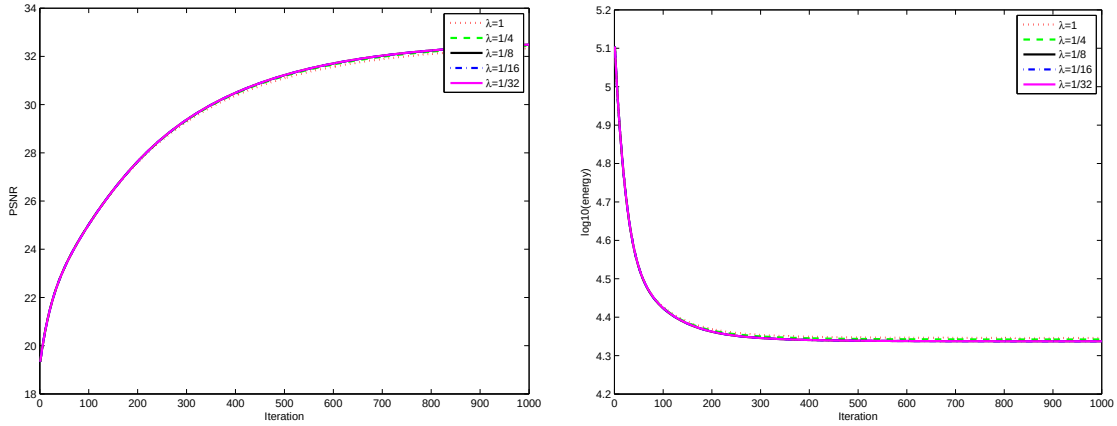
[4] Bonettini S and Ruggiero V 2012 On the convergence of primal-dual hybrid gradient algorithms for total variation image restoration *J. Math. Imaging Vis.* to appear

Figure 5.4: PSNR and energy *versus* iterations for different parameters in CT reconstruction. (a) and (b) are PSNR and energy *versus* iterations for $\kappa = 0.5, 0.1, 0.01, 0.001, 0$ ($\lambda = 1/8$, $\gamma = 1.3$). (c) and (d) are PSNR and energy *versus* iterations for $\lambda = 1, 1/4, 1/8, 1/16, 1/32$ ($\kappa = 0$, $\gamma = 1.3$). (e) and (f) are PSNR and energy *versus* iterations for $\gamma = 0.4, 0.7, 1, 1.2, 1.3$ ($\kappa = 0$, $\lambda = 1/8$)



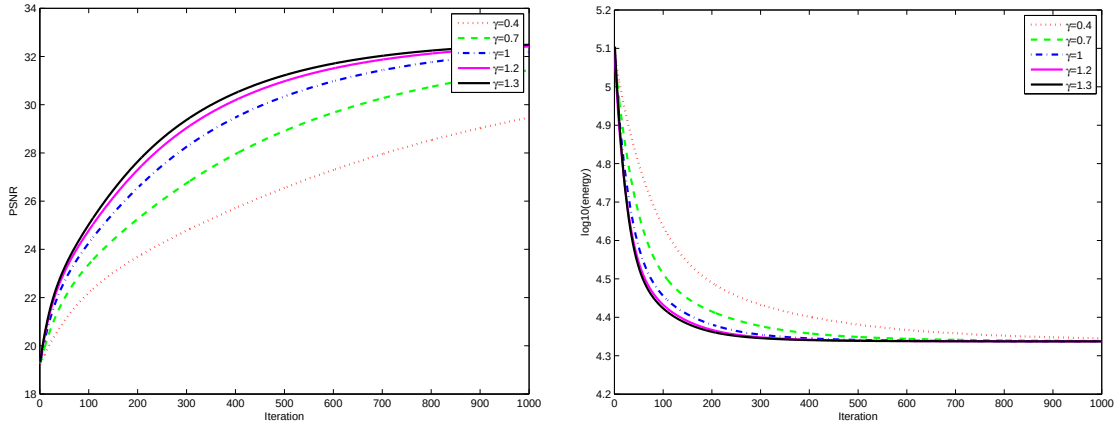
(a)

(b)



(c)

(d)



(e)

(f)

Figure 5.5: A tomographic reconstruction example for a 128×128 image, with 50 projections.

Original

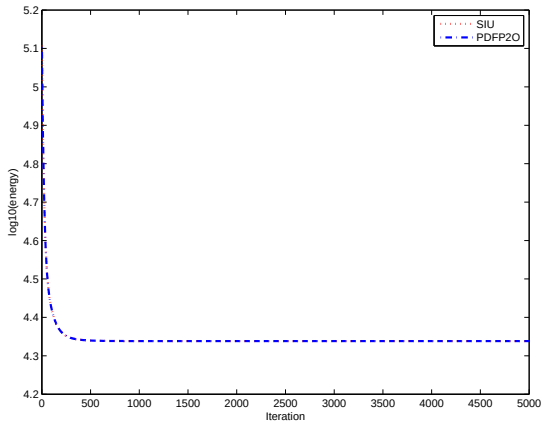
FBP

SIU, PSNR=32.7182

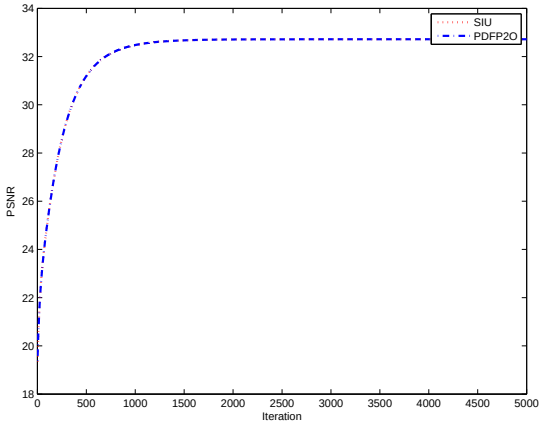
PDFP^{2}O, PSNR=32.7182

Figure 5.6: The comparisons of the performances of energy and PSNR with iterations respectively for SIU and PDFP^{2}O.

Energy vs. Iteration(5000)



PSNR vs. Iteration(5000)



[5] Boyd S, Parikh N, Chu E, Peleato B and Eckstein J 2010 Distributed optimization and statistical kearning via the alternating direction method of multipliers *Foundations and Trends in Machine Learning* 3 1-122

[6] Cai J-F, Osher S and Shen Z 2009 Linearized Bregman iterations for compressed sensing *Math. Comp.* 78 1515-36

[7] Cai J-F, Osher S and Shen Z 2009 Convergence of the Linearized Bregman Iteration for l_1-norm minimization *Math. Comp.* 78 2127-36

[8] Chambolle A 2004 An algorithm for total variation minimization and applications *J. Math. Imaging Vis.* 20 89-97

[9] Chambolle A and Pock T 2011 A first-order primal-dual algorithm for convex problems with applications to imaging *J. Math. Imaging Vis.* 40 120-45

[10] Chen D-Q, Zhang H and Cheng L-Z 2012 A fast fixed-point algorithm for total variation deblurring and segmentation *J. Math. Imaging Vis.* 43 167-79

[11] Chen G and Teboulle M 1994 A proximal-based decomposition method for convex minimization problems *Mathematical Programming* 64 81-101

[12] Chen Y, Hager W, Huang F, Phan D, Ye X and Yin W 2012 Fast Algorithms for Image Reconstruction with Application to Partially Parallel MR Imaging *SIAM J. Imag. Sci.* 5 90-118

[13] Combettes P L and Wajs V R 2005 Signal recovery by proximal forward-backward splitting *Multiscale Model. Simul.* 4 1168-200

[14] Eckstein J and Bertsekas D 1992 On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators *Mathematical Programming* 55 293-318

[15] Esser E 2009 Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split Bregman *UCLA CAM Report* [09-31]

[16] Esser E, Zhang X and Chan T F 2010 A General Framework for a Class of First Order Primal-Dual Algorithms for Convex Optimization in Imaging Science *SIAM J. Imag. Sci.* 3 1015-46.

[17] Goldstein T, O'Donoghue B and Setzer S 2012 Fast Alternating Direction Methods *UCLA CAM Report* [12-35]

[18] Goldstein T and Osher S 2009 The split Bregman method for l^1 regularized problems *SIAM J. Imag. Sci.* 2 323-43

[19] He B and Yuan X 2012 Convergence Analysis of Primal-Dual Algorithms for a Saddle-Point Problem: From Contraction Perspective *SIAM J. Imag. Sci.* 5 119-49.

[20] Lions P L and Mercier B 1979 Splitting algorithms for the sum of two nonlinear operator *SIAM J. Numer. Anal.* 16 964-79.

[21] Micchelli C A, Shen L and Xu Y 2011 Proximity algorithms for image models: denoising *Inverse Problems* 27 45009-38

[22] Michelli C A, Shen L and Xu Y 2011 Proximity algorithms for the L_1/TV image denoising model *Advances in Computational Mathematics*

[23] Moreau J-J 1962 Fonctions convexes duales et points proximaux dans un espace hilbertien *C. R. Acad. Sci., Paris Sér. A Math* 255 2897-99

[24] Nesterov Y 1983 A method of solving a convex programming problem with convergence rate O(1/k2) *Soviet Mathematics Doklady* 27 372-6

[25] Opial Z 1967 Weak convergence of the sequence of successive approximations for nonexpansive mappings *Bull. Am. Math. Soc.* 73 591-7.

[26] Osher S, Mao Y, Dong B and Yin W 2010 Fast linearized Bregman iteration for compressed sensing and sparse denoising *Communications in Mathematical Sciences* 8 93-111

[27] Pock T and Chambolle A 2011 Diagonal preconditioning for first order primal-dual algorithms in convex optimization *IEEE Int. Conf. Computer Vision (ICCV)* 1762-9

[28] Pssty G B 1979 Ergodic convergence to a zero of the sum of monotone operators in Hilbert space *J. Math. Anal. Appl.* 72 383-90

[29] Rockafellar R T 1970 *Convex Analysis* (Princeton, NJ: Princeton University Press)

[30] Rudin L I, Osher S and Fatemi E 1992 Nonlinear total variation based noise removal algorithms *Physica D.* 60 259-68

[31] Setzer S 2009 Split Bregman Algorithm, Douglas-Rachford Splitting and Frame Shrinkage *Scale Space and Variational Methods in Computer Vision* 5567 464-76

[32] Tseng P 2008 On accelerated proximal gradient methods for convex-concave optimization *Preprint*

[33] Tseng P 2010 Approximation accuracy, gradient methods, and error bound for structured convex optimization *Mathematical Programming* 125 263-95

[34] Yin W, Osher S, Goldfarb D and Darbon J 2008 Bregman iterative algorithms for l_1 minimization with applications to compressed sensing *SIAM J. Imag. Sci.* 1 143-68

[35] Yin W and Peng 2012 On the global linear convergence of Alternating direction methods *Private communication*

[36] Zhang X, Burger M, Bresson X and Osher S 2010 Bregmanized Nonlocal Regularization for Deconvolution and Sparse Reconstruction *SIAM J. Imag. Sci.* 3 253-76

[37] Zhang X, Burger M and Osher S 2011 A unified primal-dual algorithm framework based on Bregman iteration *J. Sci. Comput.* 46 20-46

[38] Zhu M and Chan T F 2008 An Efficient Primal-Dual Hybrid Gradient Algorithm for Total Variation Image Restoration *UCLA CAM Report* [08-34]