VNU Journal of Science, Mathematics - Physics 27 (2011) 67-75

On strongly convergent parallel proximal point algorithms

Pham Ky Anh*, Cao Van Chung

Faculty of Mathematics, Mechanics and Informatics
VNU University of Science, 334 Nguyen Trai, Hanoi, Vietnam

Received 30 March 2011

Abstract. This paper is concerned with two parallel proximal point algorithms for solving a system of ill-posed equations involving monotone operators. They are parallel versions of the projection-proximal point method proposed by Solodov and Svaiter and the regularization-proximal point method introduced by Ryazantseva, respectively. The convergence analysis of both methods has been investigated. The paper is completed by some numerical experiments.
Keywork: monotone operator, proximal point method, iterative regularization method, parallel computation.
2000 Mathematics Subject Classifications: 47H17, 47J06, 47J25, 65J15, 65J20, 65Y05.

1. Introduction

Various problems of science and engineering, such as the convex feasibility problems with applications in optimization theory, image processing, radiation therapy treatment planning, etc... (see [1]), or parameter identification problems with multi-sources [2], can be reduced to finding a solution of a simultaneous system of possibly nonlinear operator equations.

For solving a maximal monotone operator inclusion, Rockafellar [3] proposed the proximal point algorithm, which is in general only weakly convergent [4]. Solodov and Svaiter [5] combined the proximal point algorithm with a simple projection step onto intersection of appropriately constructed halfspaces to get the strong convergence. Later on, Ryazantseva [6, 7] proposed a strongly convergent algorithm combining the proximal point method and Lavrentiev regularization technique.

The aim of this article is to apply the projection-proximal point and the regularization-proximal point algorithms in *a parallel way* to the following consistent system of operator equations:

$$A_i(x) = 0, \quad i = \overline{1, N}, \tag{1}$$

where H is a real Hilbert space and $A_i : H \to H$ are continuous monotone operators, i.e.,

$$\langle A_i(x) - A_i(y), x - y \rangle \geq 0, \qquad \forall x, y \in H.$$

The rest of the paper is organized as follows. In Section 2 we study a parallel version of the projection-proximal point algorithm, which becomes the Solodov - Svaiter's method if the number of equations $N = 1$. Section 3 deals with a parallel regularization-proximal point method, which can be regarded as a parallel implicit iterative regularization method considered in [8, 9]. The convergence of the method

* Corresponding author. E-mail: anhpk@vnu.edu.vn

is investigated in the noisy data case. Finally, in Section 4, two parallel algorithms are tested by some model problems.

2. Parallel projection-proximal point method

We begin this section by recalling some notations and results in [5].

Theorem 2.1. *Let C be any nonempty closed convex set in H, for $x, y \in H$ and $z \in C$. Then the orthogonal projector P_C from H onto C satisfies the following relations.*

$$< x - P_C(x), z - P_C(x) > \leq 0; \tag{2}$$

$$\|P_C(x) - P_C(y)\|^2 \leq \|x - y\|^2 - \|(P_C(x) - x) - (P_C(y) - y)\|^2. \tag{3}$$

Lemma 2.2. *Let $A : H \to H$ be a monotone operator, $x \in H$, $\mu > 0$, $\sigma \in [0, 1)$ and suppose that $y \in H$ satisfies*

$$A(y) + \mu(y - x) + e = 0, \quad \textit{where} \quad \|e\| \leq \sigma \max\{\|A(y)\|, \mu\|x - y\|\}.$$

Then we have

$$< x - y, A(y) > \geq \sigma \max\{\mu\|x - y\|^2, \|A(y)\|^2/\mu\} \geq (1 - \sigma)\|A(y)\|\|x - y\|.$$

Define a half-space $H_y = \{z \in H \big| < z - y, A(y) > \leq 0\}$, then the following four statements are equivalent:

$$(i) \quad x \in H_y; \quad (ii) \quad y = x; \quad (iii) \quad A(y) = 0; \quad (iv) \quad A(x) = 0.$$

Furthermore,

$$\|P_{H_y}(x) - x\| \geq (1 - \sigma) \max\{\|x - y\|, \|A(y)\|/\mu\}. \tag{4}$$

For solving system (1) with a nonempty solution set

$$S = \{z \in H| \quad A_i(z) = 0, \quad i = \overline{1, N}\} \neq \emptyset$$

and A_i are continuous monotone operators, we implement the following parallel algorithm on a computing cluster with N processors.

Algorithm 2.1. *Let $x_0 \in H$ be an arbitrary initial point, $\overline{\mu} > 0$ and $\sigma \in [0, 1)$.*

- *At iteration $k \geq 0$, having x_k, we compute (in parallel) solutions $y_k^i \in H$ of equations*

$$A_i(y_k^i) + \mu_k^i(y_k^i - x_k) + e_k^i = 0, \quad i = \overline{1, N}, \tag{5}$$

where $\mu_k^i \in (0, \overline{\mu})$, $\|e_k^i\| \leq \sigma \max\{\|A_i(y_k^i)\|, \mu_k^i\|x_k - y_k^i\|\}$.

- *Define (in parallel) projections from x_k onto half-spaces*

$$H_k^i = \{z \in H \big| < z - y_k^i, A_i(y_k^i) > \leq 0\}$$

and find an optimal index j_k $(1 \leq j_k \leq N)$, such that

$$\|x_k - P_{H_k^{j_k}}(x_k)\| = \max_{i=\overline{1,N}}\{\|x_k - P_{H_k^i}(x_k)\|\}.$$

- *Compute*

$$x_{k+1} = P_{H_k^{j_k} \cap W_k}(x_0), \tag{6}$$

where $W_k = \{z \in H \mid < z - x_k, x_0 - x_k > \leq 0\}$.

If $x_{k+1} = x_k$ *then stop. Else, set* $k := k+1$ *and repeat.*

Since A_i is monotone and $\mu_k^i > 0$, each subproblem (5) is well-posed, hence it has a unique solution y_k^i. At each iteration k, if $x_k \in H_k^i$ then $P_{H_k^i}(x_k) = x_k$ and $\|x_k - P_{H_k^i}(x_k)\| = 0$. Otherwise, we have

$$P_{H_k^i}(x_k) = x_k - \frac{< A_i(y_k^i), x_k - y_k^i >}{\|A_i(y_k^i)\|^2} A_i(y_k^i) \quad \text{and} \quad \|x_k - P_{H_k^i}(x_k)\| = \frac{| < A_i(y_k^i), x_k - y_k^i > |}{\|A_i(y_k^i)\|}.$$

Clearly, the computation of the optimal index j_k at iteration k of Algorithm does not require much additional cost.

The convergence of Algorithm can be established by the technique introduced in [5].

Lemma 2.3. *If the Algorithm terminates at a finite iteration* $k+1$, *then* x_k *is a solution of system* (1).

Proof. If the Algorithm terminates at a finite iteration $k+1$, then we have $x_{k+1} = P_{H_k^{j_k} \cap W_k}(x_0) \equiv x_k$. It follows $x_k \in H_k^{j_k}$ and therefore $\|x_k - P_{H_k^{j_k}}(x_k)\| = 0$. By the definition of j_k, we have $\|x_k - P_{H_k^i}(x_k)\| = 0$ for all $i = \overline{1, N}$. Now applying Lemma 2.2 to each equation $A_i(y_k^i) + \mu_k^i(y_k^i - x_k) + e_k^i = 0$ with respect to $x = x_k$, $y = y_k^i$, we have

$$\|P_{H_k^i}(x_k) - x_k\| \geq (1 - \sigma) \max\{\|x_k - y_k^i\|, \|A_i(y_k^i)\|/\mu_k^i\} \quad \text{for all} \quad i = \overline{1, N}.$$

Hence, $A_i(y_k^i) = 0$ and $y_k^i \equiv x_k$ for all $i = \overline{1, N}$, or x_k is a solution of system (1).

In what follows, assuming that Algorithm generates an infinite sequence x_k, we will show that knowing the $k-$th iterate x_k we can define the next one x_{k+1}. For a chosen initial iterate $x_0 \in H$ we define the set

$$U(x_0) = \{x \in H \mid \quad \forall z \in S, \quad < z - x, x_0 - x > \leq 0\}.$$

Clearly, $x_0 \in U(x_0)$.

Lemma 2.4. *Suppose that at iteration k-th of algorithm we have* $x_k \in U(x_0)$, *then*

i. $S \subset (\cap_{i=1}^N H_k^i) \cap W_k \subset H_k^{j_k} \cap W_k$.

ii. x_{k+1} *from* (6) *is well-defined and* $x_{k+1} \in U(x_0)$.

iii. $\|x_{k+1} - x_0\| \leq \|P_S(x_0) - x_0\|$ *for all* $k \in \mathbb{N}$, *and therefore* $\{x_k\}$ *is bounded.*

Proof. From the monotonicity of A_i, for any $z \in S$ we have

$$< A_i(y_k^i), z - y_n^i >= - < A_i(y_k^i) - A_i(z), y_n^i - z > \leq 0, \quad i = \overline{1, N}.$$

Then $z \in (\cap_{i=1}^N H_k^i)$, and hence $S \subset (\cap_{i=1}^N H_k^i)$. Since $x_k \in U(x_0)$, it follows $< z - x_k, x_0 - x_k > \leq 0$ for all $z \in S$. Therefore, $z \in W_k$ and $S \subset W_k$. Thus, $S \subset (\cap_{i=1}^N H_k^i) \cap W_k \subset H_k^{j_k} \cap W_k$, and the assumption $S \neq \emptyset$ implies that $H_k^{j_k} \cap W_k \neq \emptyset$. Hence $x_{k+1} = P_{H_k^{j_k} \cap W_k}(x_0)$ is well-defined.

Since x_{k+1} is the projection of x_0 onto $H_k^{j_k} \cap W_k$, from (2) we have $< z - x_{k+1}, x_0 - x_{k+1} > \leq 0$ for all $z \in H_k^{j_k} \cap W_k$. The inclusion $S \subset H_k^{j_k} \cap W_k$ and the last inequality ensure that $< z - x_{k+1}, x_0 - x_{k+1} > \leq 0$ for all $z \in S$, therefore $x_{k+1} \in U(x_0)$.

From (6), we also have $\|x_{k+1} - x_0\| \leq \|z - x_0\|$ for all $z \in H_k^{j_k} \cap W_k$. Taking into account the inclusion $S \subset H_k^{j_k} \cap W_k$, $\forall z \in S$, we have $\|x_{k+1} - x_0\| \leq \|z - x_0\|$, i.e., $\|x_{k+1} - x_0\| \leq \|P_S(x_0) - x_0\|$, which implies the boundedness of the sequence $\{x_k\}$.

By Lemma 2.4, starting from $x_0 \in U(x_0)$, we have $x_k \in U(x_0)$ for $k = 0, 1, 2, \ldots$

Lemma 2.5. *Suppose the Algorithm reaches an iteration $k + 1$, then we have*

$$\|x_{k+1} - x_0\|^2 \geq \|x_k - x_0\|^2 + \|x_{k+1} - x_k\|^2, \tag{7}$$

$$\|x_{k+1} - x_k\| \geq (1 - \sigma) \max_{i=\overline{1,N}} \{\|y_k^i - x_k\|, \|A_i(y_k^i)\|/\mu_k^i\}. \tag{8}$$

Proof. From the definition of W_k, it follows that $x_k = P_{W_k}(x_0)$. Applying (3) with respect to $C = W_k$, $x = x_{k+1}$ and $y = x_0$, we have

$$\|P_{W_k}(x_{k+1}) - P_{W_k}(x_0)\|^2 \leq \|x_{k+1} - x_0\|^2 - \|x_{k+1} - P_{W_k}(x_{k+1}) - (x_0 - P_{W_k}(x_0))\|^2.$$

Now observing that $P_{W_k}(x_{k+1}) = x_{k+1}$, since $x_{k+1} \in W_k$, and $P_{W_k}(x_0) = x_k$, we get (7). On the other-hand, since $x_{k+1} \in H_k^{j_k}$, it follows

$$\|x_k - x_{k+1}\| \geq \|x_k - P_{H_k^{j_k}}(x_k)\| \geq \max_{i=\overline{1,N}} \|x_k - P_{H_k^i}(x_k)\|. \tag{9}$$

Using the last inequality and applying (4) with respect to $H_y := H_k^i$, $A := A_i$, $x := x_0$, $\mu := \mu_k^i$ and $y := y_k^i$, we have $\|x_k - P_{H_k^i}(x_k)\| \geq (1 - \sigma) \max\{\|y_k^i - x_k\|, \|A_i(y_k^i)\|/\mu_k^i\}$. Finally, from the last relation and (9) we come to the estimate (8).

Theorem 2.1. *Let $\{x_k\}$ be the infinite sequence generated by Algorithm , then*

$$\lim_{k\to\infty} x_k = P_S(x_0).$$

Proof. Using (7) consecutively, we have

$$\|x_{k+1} - x_0\|^2 \geq \|x_k - x_0\|^2 + \|x_{k+1} - x_k\|^2 \geq \sum_{l=0}^{k-1} \|x_{l+1} - x_l\|^2. \tag{10}$$

From item (iii) of Lemma 2.4 and (10), we have $\sum_{l=0}^{\infty} \|x_{l+1} - x_l\|^2 \leq \|P_S(x_0) - x_0\|^2 < \infty$, therefore $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$. Using (8) and taking into account that $\mu_k^i \leq \overline{\mu}$, we also have $\lim_{k\to\infty} \|y_k^i - x_k\| = 0$, and $\lim_{k\to\infty} \|A_i(y_k^i)\| = 0$, for all $i = 1, 2, \ldots, N$.

Since $\{x_k\}$ is bounded, it is relatively weakly compact. Let $\{x_{k_m}\}$ be an arbitrary weakly convergent subsequence of the bounded sequence $\{x_k\}$ and $x_{k_m} \rightharpoonup \tilde{x}$ as $m \to \infty$. Clearly, $y_{k_m}^i \rightharpoonup \tilde{x}$ as $m \to \infty$. By the monotonicity of A_i, for each $i = 1, 2, ..., N$ and any $z \in H$, we have

$$0 \leq < z - y_{k_m}^i, A_i(z) - A_i(y_{k_m}^i) > = < z - y_{k_m}^i, A_i(z) > - < z - y_{k_m}^i, A_i(y_{k_m}^i) > .$$

Passing to the limit as $m \to \infty$ and taking into account $y_{k_m}^i \rightharpoonup \tilde{x}$ and $A_i(y_{k_m}^i) \to 0$, we find

$$< z - \tilde{x}, A_i(z) > \geq 0 \qquad \forall z \in H, \quad i = 1, 2, \ldots, N.$$

Now from the maximal monotonicity of A_i it follows $A_i(\tilde{x}) = 0$ (see [7]), $i = \overline{1, N}$, i.e., $\tilde{x} \in S$. Using the relation $\|x_k - x_0\| \le \|P_S(x_0) - x_0\|$ for all k, we get

$$\begin{aligned}\|x_{k_m} - P_S(x_0)\|^2 &= \|x_{k_m} - x_0 - (P_S(x_0) - x_0)\|^2 \\ &= \|x_{k_m} - x_0\|^2 + \|P_S(x_0) - x_0\|^2 - 2 < x_{k_m} - x_0, P_S(x_0) - x_0 > \\ &\le 2\|P_S(x_0) - x_0\|^2 - 2 < x_{k_m} - x_0, P_S(x_0) - x_0 > .\end{aligned}$$

Hence,

$$\lim_{m\to\infty} \sup \|x_{k_m} - P_S(x_0)\|^2 \le 2\Big(\|P_S(x_0) - x_0\|^2 - < \tilde{x} - x_0, P_S(x_0) - x_0 > \Big). \tag{11}$$

Applying (2) with respect to $C := S$, $x = x_0$ and $z := \tilde{x} \in S$, we have

$$< x_0 - P_S(x_0), \tilde{x} - P_S(x_0) > = \|x_0 - P_S(x_0)\|^2 - < \tilde{x} - x_0, P_S(x_0) - x_0 > \le 0.$$

Combining the last inequality with (11), we find $\lim_{k_m\to\infty} \|x_{k_m} - P_S(x_0)\| = 0$ or $x_{k_m} \to P_S(x_0)$ as $m \to \infty$. Moreover, we also have $\tilde{x} \equiv P_S(x_0)$. Thus, $P_S(x_0)$ is the unique weak accumulation point of $\{x_k\}$. Clearly, every weakly convergent subsequence of $\{x_k\}$ strongly converges to $P_S(x_0)$, therefore $x_k \to P_S(x_0)$ as $k \to \infty$.

3. Parallel regularization-proximal point method

In this section we consider system (1) with $A_i(x) := F_i(x) - f_i$, where $F_i : H \to H$, $(i = \overline{1, N})$ are supposed to be c^{-1}–inverse-strongly monotone operator (see [10]), i.e.,

$$< F_i(x) - F_i(y), x - y > \ge c^{-1}\|F_i(x) - F_i(y)\|^2, \qquad \forall x, y \in H, \quad c > 0.$$

We assume as in Section 2 that the solution set $S \subset H$ of (1) is not empty, hence S is convex and closed. Furthermore, suppose that $0 \notin S$.

Let $F(x) = \sum_{i=1}^{N} F_i(x)$, $f = \sum_{i=1}^{N} f_i$ and $A(x) := F(x) - f$ for all $x \in H$. Suppose that instead of exact data $\{F_i, f_i\}$, we are given only noisy ones $\{F_{n,i}, f_{n,i}\}$, such that

$$\|F_{n,i}(x) - F_i(x)\| \le h_n g(\|x\|), \quad \forall x \in H, \quad \|f_{n,i} - f_i\| \le \delta_n, \quad n = 1, 2, ...,$$

where $\delta_n > 0$, $h_n > 0$ are noise levels and $g : \mathbb{R}^+ \to \mathbb{R}^+$ is a positive nondecreasing function.

We put $A_{n,i}(x) := F_{n,i}(x) - f_{n,i}$, $A_n(x) = \sum_{i=1}^{N} A_{n,i}(x)$ and suppose that the operators $F_{n,i}$: $H \to H$ are continuous and monotone. Combining the parallel splitting up technique [11] with the regularization-proximal point method [6] for the equation $A_n(x) = 0$, we come to the following parallel regularization proximal point (PRPXP) method

$$A_{n,i}(z_n^i) + \left(\frac{\alpha_n}{N} + \frac{1}{c_n} + \tilde{\gamma_n}\right) z_n^i = \left(\frac{1}{c_n} + \tilde{\gamma_n}\right) z_n, \qquad i = 1, 2, \ldots, N, \tag{12}$$

$$z_{n+1} = \frac{1}{N}\sum_{i=1}^{N} z_n^i, \qquad n = 0, 1, 2, \ldots, \tag{13}$$

Clearly, the main computational task (12) can be performed simultaneously by N parallel processors. With notation $\gamma_n := \dfrac{1}{c_n} + \tilde{\gamma_n}$, the PRPXP method (12)-(13) becomes a parallel implicit iterative regularization method (PIIRM) proposed in [8], whose convergence has been studied in the noise-free case only.

Denoting $\gamma_n := \tilde{\gamma}_n + \frac{1}{c_n}$, where $\tilde{\gamma}_n$ and c_n are mentioned in (12)-(13), and $x^\dagger$ the minimal - norm solution of the system $A_i(x) = F_i(x) - f_i = 0$ $(i = \overline{1, N})$, we have following convergence result.

Theorem 3.1. *Let α_n and γ_n be two sequences of positive numbers, such that $\alpha_n \searrow 0$, $\gamma_n \nearrow +\infty$ as $n \to +\infty$ and suppose that the following conditions are satisfied for all $n \in \mathbb{N}$ and some constant $m_1 > 0$*

$$\frac{\gamma_n(\alpha_n - \alpha_{n+1})}{\alpha_n^3} \leq \frac{m_1\gamma_0}{\alpha_0^2}; \qquad \gamma_n\alpha_n^2 \geq \gamma_0\alpha_0^2; \qquad \frac{h_n g(\|x^\dagger\|) + \delta_n}{h_0 g(\|x^\dagger\|) + \delta_0} \leq \frac{\sqrt{\alpha_n}}{\gamma_n}.$$

Further, we assume that $(1 - 4m_1 + m_1^2)\alpha_0 > 4m_1 N\gamma_0$, $\alpha_0\gamma_0 \geq N$ and $\|x^\dagger\|^2 \leq l\alpha_0^2$, where

$$l := \frac{2(2N\gamma_0 + \alpha_0)}{\gamma_0[(1 - 4m_1 + c_1^2)\alpha_0 - 4m_1 N\gamma_0]}\left\{\left[\frac{2c}{\gamma_0\alpha_0} + \frac{1}{N^2\gamma_0} + \frac{c_1\gamma_0(N\gamma_0 + \alpha_0)}{\alpha_0^3}\right]\|x^\dagger\|^2 + \frac{\left(h_0 g(\|x^\dagger\|) + \delta_0\right)^2}{2\gamma_0\alpha_0}\right\}.$$

Then starting from $z_0 = 0$, the sequence z_n converges to $x^\dagger$.

Although the proof of this theorem is complicated, it follows the same line as the proof of Theorem 2.1 in [8], therefore it will be omited.

Remark 3.1. The sequences $\alpha_n = \alpha_0(1+n)^{-p}$; $\gamma_n = \gamma_0(1+n)^{1/2}$, where $0 < p \leq \frac{1}{4}$ and the constants $c_1 = \frac{1}{4}$, $\gamma_0 = \frac{N}{\alpha_0}$ and $\alpha_0 > 4N$ satisfy all the requirements in Theorem 3.1.

Remark 3.2. If the operators $F_i(x)$ are free of noise, i.e., $h_n \equiv 0$ and the noise levels δ_n do not satisfy a-priori conditions in Theorem 3.1, then method (12)-(13) may not converge to the minimal norm solution $x^\dagger$ of (1). However, we can choose an appropriate stopping number of iterates $n = n_\delta$ such that the sequence z_{n_δ} still gives stable approximations for $x^\dagger$. Moreover, $z_{n_\delta} \to x^\dagger$ as $\delta \to 0$. This problem will not be discussed here due to lack of space.

4. Numerical experiments

To test the described above parallel proximal point methods we consider the system of linear first kind Fredholm integral equations given in [8]:

$$(A_i x)(t) := \int_a^b K_i(t,s)x(s)ds - f_i(t) = 0 \qquad i = 1, 2, \ldots, N, \tag{14}$$

where $N = 4$; $[a, b] \equiv [0, 1]$ and the kernels $K_1(t,s) = \frac{ts}{3}$; $K_2(t,s) = \frac{1}{3} + \frac{t+s}{2} + ts$;

$$K_3(t,s) = \begin{cases} s(1-t) & s \leq t; \\ t(1-s) & t \leq s \end{cases}; \qquad K_4(t,s) = \begin{cases} \frac{(t-s)^3}{3} - \frac{t+s}{2} + ts + \frac{1}{3} & s \leq t; \\ \frac{(s-t)^3}{3} - \frac{t+s}{2} + ts + \frac{1}{3} & t \leq s \end{cases}$$

It was shown in [8] that A_i, $i = 1, 2, ..., N$ are inverse-strongly monotone operators. In particular, A_i are Lipschitz continuous, i.e., $\|A_i(x) - A_i(y)\| \leq L\|x - y\|$ for all $x, y \in H$, with $L \geq \max_{i=\overline{1,N}}\{\left(\int_0^1\int_0^1 K_i^2(t,s)ds\right)^{1/2}\}$. For an arbitrary fixed constant $q \in (0, 1)$, we choose μ_k^i such that $0 < \underline{\mu} \leq \mu_k^i \leq \overline{\mu}$, where $\underline{\mu}$ satisfies $L/\underline{\mu} \leq q < 1$.

The integrals in the left-hand sides of (14) are discretized by the rectangle rule. The programs are written in C and executed on a Linux cluster 1350 with 8 computing nodes of 51.2 GFlops. Each node contains 2 Intel Xeon dual core 3.2 GHz, 2GB Ram. The notations used in this section are as follows.

TOL	Tolerances
M	Number of equal-length subdivisions of $[0, 1]$
$n_{\max}$	Total number of iterations
T	Time of the parallel execution on 4 CPUs taken in seconds
PRPXPM	Parallel regularization proximal point method
PPPXPM	Parallel projection proximal point method
!NS&DE	Method is not stable and explosively divergent.

Firstly, we consider two methods PRPXP and PPPXP in a free noise case. Then, PRPXP method is equivalent to PIIR method [8]. We choose the initial approximation $x_0 \equiv 0$. the parameters $\alpha_n = \dfrac{1}{8(n+1)^{2/5}}$, $\gamma_n = \dfrac{\sqrt{n+1}}{8}$ for PRPXP method and $\mu_k^i \equiv 3.5$, $\sigma = 0.5$ for Algorithm . The following right-hand sides

$$f_1(t) = \frac{t}{9}; \quad f_2(t) = \frac{7t+4}{12}; \quad f_3(t) = \frac{t-t^3}{6}; \quad f_4(t) = \frac{t^5 - 5t^3 + 10t^2 - 5t + 2}{30} \tag{15}$$

and

$$\begin{aligned} f_1(t) &= -\frac{t}{6\pi}; \quad f_2(t) = -\frac{2t+1}{4\pi}; \quad f_3(t) = \frac{\sin(2t\pi)}{4\pi^2}; \\ f_4(t) &= \frac{8\pi^2t^3 - 12\pi^2t^2 - 12t + 2\pi^2 + 6}{24\pi^3} + \frac{\sin(2\pi t)}{4\pi^4}, \end{aligned} \tag{16}$$

corresponding to exact solutions $x_{e1}(t) = t$ and $x_{e2}(t) = \sin(2\pi t)$, respectively, are given in [8]. Performance results for a small number of iterations are showed in the following tables.

Table 4.1. Free noise cases and small number of iterations

right-hand sides $f_i; i = 1,2,3,4$	M	$n_{\max}$	PRPXPM T	PRPXPM TOL	PPPXPM T	PPPXPM TOL
(15) - $x_e(t) = t$	128	500	0.98	0.00636	0.51	0.00147
		750	1.42	0.00492	0.97	0.00115
		1500	2.77	0.00399	1.42	0.00098
	256	500	4.50	0.00518	1.04	0.00192
		1000	8.65	0.00311	2.10	0.00124
(15) - $x_e(t) = \sin(2\pi t)$	128	500	0.93	0.00651	0.48	0.00123
		1000	1.89	0.00519	1.01	0.00086
	256	500	4.38	0.00557	0.99	0.00178
		1000	8.61	0.00323	1.97	0.00107

Table 4.1 shows that in a free noise case, if the number of iterations is small, then the PRPXP method is more time consuming than PPPXP method. For a fixed number of iterations, the PPPXP

method is also more accurate than PRPXP method. The next table shows the results in a free noise case, when the number of iterations is large.

Table 4.2. Free noise cases, large number of iterations, and $x_{e2}(t) = \sin(2\pi t)$

M	PRPXPM			PPPXPM		
	TOL	$n_{\max}$	T	TOL	$n_{\max}$	T
128	0.001051	15000	26.87	0.000475	15000	14.02
	0.000835	50000	80.02	0.000397	50000	41.17
	0.000759	100000	159.29	0.000401	100000	80.51
256	0.000285	553153	4701.12	0.000285	23427	45.87
	0.000230	1173089	9969.18	0.000230	135311	265.09
	0.000200	2798307	23781.01	0.000200	!NS&DE	!NS&DE

From Table 4.2 we observe that the PRPXP method may be more time consuming than PPPXP method, but it is always stable and convergent. On the other hand, due to the dicretization and round-off errors, the PPPXP method may be unstable whenever the number of iterations is large. Moreover, this method may give an unsatisfactory result within a given small tolerance.

Now we consider the noisy case with α_n and γ_n are chosen as in Remark 3.1. For the sake of simplicity, we use $F_{n,i}(x) = F_i(x) + \dfrac{\sqrt{\alpha_n}\rho_n(t)}{\gamma_n}x$ and $f_{n,i} = f_i + \dfrac{\sqrt{\alpha_n}\rho_n(t)}{\gamma_n}$, where $\rho_n(t) := 0.25\varrho_n t$ and $\varrho_n \in [0;1]$ are normally distributed random numbers with zero mean. In this experiment, we set $M = 256$.
The Table 4.3 shows that in all cases, the PRPXP method is stable and convergent. But it may be more time consuming than PPPXP method. On the other hand, due to the error of data, the PPPXP method may be unstable and divergent.

Table 4.3. Noisy data cases

right-hand sides f_i; $i = 1, 2, 3, 4$	PRPPM			PPPPM		
	$n_{\max}$	T	TOL	$n_{\max}$	T	TOL
(15) - (w.r.t $x_e(t) = t$)	1000	8.51	0.00761	1000	2.13	0.00157
	20000	166.06	0.00505	20000	45.00	0.00075
	543875	4615.5	0.00105	2437	4.35	0.00105
	8752118	68112.6	0.00050	!NS&DE	!NS&DE	0.00050
(16) - (w.r.t $x_e(t) = \sin(2\pi t)$)	1000	8.67	0.00693	1000	2.07	0.00233
	5000	44.12	0.00575	5000	10.61	0.00098
	20000	174.15	0.00545	20000	42.92	0.00104
	635224	5481.2	0.00100	13047	26.33	0.00100
	2873115	21924.7	0.00075	!NS&DE	!NS&DE	0.00075

5. Conclusion

In this note two parallel versions of the proximal point method for solving a system of ill-posed nonlinear operator equations are studied. Based on parallel computation we can reduce the overall computational effort without imposing extra conditions on the nonlinearity of the operators. Experiments show that the PRPXP method is more time consuming but is much stabler than the

PPPXP method, especially in the noisy data case. Other parallel methods for ill-posed problems can be found in [12, 13].

Acknowledgements. The work is supported by the Vietnam National Foundation for Science and Technology Development under the contract 101.02.42.09.

References

[1] Y. Censor, *On sequential and parallel projection algorithms for feasibility and optimization*, Proc. SPIE, vol. 4553, SPIE- The Society for Optical Engineering, Bellingham, WA, USA, 2001, 1

[2] M. Burger and B. Kaltenbacher, Regularizing Newton-Kaczmarz methods for nonlinear ill-posed problems, *SIAM Numer. Anal.*, 44 (2006) 153.

[3] R.T. Rockafellar, Monotone operators and proimal point algorithm,*SIAM J. Contr. Optim.* 14(1976) 877.

[4] O. Guler, On the convergence of proximal point algorithm for convex minimization,*SIAM J. Contr. Optim.* 29 (1991) 403.

[5] M.V. Solodov, B.F. Svaiter, Forcing strong convergence of proximal point iterations in Hilbert space, *Math. Progr.* 87 (2000) 189.

[6] I.P. Ryazantseva, Regularizing proximal point algorithm for nonlinear equations of monotone type,*J. Comput. Math. Math. Phys.* 42(9) (2002)1295 (in Russian).

[7] Y. Alber and I. Ryazantseva, *Nonlinear Ill-posed Problems of Monotone Type*, Springer, 2006.

[8] P. K. Anh, C. V. Chung, Parallel iterative regularization methods for solving systems of ill-posed equations, *Appl. Math. Comput.*, 212 (2009) 542.

[9] P.K. Anh and V.T. Dung, Parallel iterative regularization algorithms for large overdetermined linear systems,*Inter. J. Comput. Meth.*, 7(4) (2010) 525.

[10] F. Liu, M. Z. Nashed, Regularization of nonlinear ill-posed variational inequalities and convergence rates, *Set-Valued Analysis*, 6 (1998) 313.

[11] T. Lü, P. Neittaanmäki, X.-C. Tai, A parallel splitting up method for partial differential equations and its application to Navier-Stokes equations,*RAIRO Math. Model. Numer. Anal.*, 26 (6) (1992) 673.

[12] P.K. Anh, C.V. Chung *Parallel regularized Newton method for nonlinear ill-posed equations*, Numer. Algor., 2011, DOI: 10.1007/s11075-011-9460-y.

[13] P.K. Anh, C.V. Chung, V.T. Dung, Cimmino methods for regularizing nonlinear ill-posed problems,*Proc. Inter. Confer. Anal. Appl. Math.*, Saigon Univ., HCM City, March, 13, 2011, 67.