Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH Open Access

A preconditioning method of the CQ algorithm for solving an extended split feasibility problem

Peiyuan Wang[1*] and Haiyun Zhou[1,2]

*Correspondence: wangpy629@163.com
[1]Department of Mathematics, Shijiazhuang Mechanical Engineering College, Shijiazhuang, 050003, China
Full list of author information is available at the end of the article

Abstract
In virtue of preconditioning technology, we propose a preconditioning CQ algorithm for an extend split feasibility problem (ESFP). Comparing with the others, the proposed algorithm can get faster convergence without considering to adjust the stepsize. The convergence is also established under mild conditions. Several extensions of the preconditioning CQ algorithm are presented. Moreover, we present an approximate variable preconditioner which does not compute the matrix inverse. Finally, some numerical experiments show the better behaviors of the proposed methods.

MSC: 47J10; 47J20; 65B05

Keywords: preconditioning method; extended split feasibility problem; relaxed method; approximate variable preconditioner

1 Introduction

The problem to find $x \in C$ with $Ax \in Q$, if such x exists, was called the split feasibility problem (SFP) by Censor and Elfving [1], where $C \in R^N$ and $Q \in R^M$ are nonempty closed convex sets, and A is an M by N matrix. This problem plays an important role in the study of signal processing, image reconstruction, and so on [2, 3]. Censor and Elfving's algorithm in [1], as well as others obtained later [4, 5] involve matrix inverses at each step. Byrne [6] presented a method called the CQ algorithm for solving the SFP that does not involve matrix inverses.

The CQ algorithm Let x^0 be arbitrary. For $k = 0, 1, \dots,$ calculate

$$x^{k+1} = P_C\bigl(x^k - \gamma A^T(I - P_Q)Ax^k\bigr), \tag{1}$$

where $\gamma \in (0, 2/L)$ and L denotes the largest eigenvalue of the matrix A^TA, I is the identical matrix. P_C and P_Q are the orthogonal projections onto C and Q, respectively.

In recent years, how to modify the CQ algorithm so that it can easily be implemented and converge faster is the hot topic. The typical modifications are as follows: Yang [7] presented a relaxed CQ algorithm for solving the SFP, then the orthogonal projections onto halfspaces C_k and Q_k can be executed exactly. Qu and Xiu [8] proposed the modified relaxed algorithm which does not need to compute the largest eigenvalue of the matrix

A^TA and can get an adaptive stepsize by adopting an Armijo-like search. The paper [9] extended the algorithm in [10] and proposed a relaxed inexact projection method for the SFP. Xu [11] extended the problem into infinite-dimensional Hilbert spaces, and modified the CQ algorithm with Mann's iteration. In [12], López *et al.* presented a variable stepsize, and improved the algorithm with a Halpern-type iteration.

However, using preconditioning technology to accelerate the CQ algorithm not only has not been taken into account, but also one will obtain a special effect. In this paper, we consider to modify the CQ algorithm from the views of fixed point and variational inequality. Combining with the appropriate preconditioner, the SFP can be transformed into an extended split feasibility problem (ESFP). Naturally, a preconditioning CQ algorithm for solving the ESFP can also solve the SFP indirectly.

The rest of the paper is organized as follows. In Section 2, we review some concepts and existing results. In Section 3, we propose a preconditioning CQ algorithm for solving ESFP and establish its convergence. Several extensions are presented in Section 4. In Section 5, we discuses the methods how to estimate the approximate inverse preconditioner. In Section 6, we report some computational results with the proposed algorithm and methods. Finally, Section 7 gives some concluding remarks.

2 Preliminaries

Our argument mainly depends on monotone operators, nonexpansive mappings, and the metric projections.

Definition 2.1 [13] Let T be a mapping from a set $C \subset R^N$ into itself. Then

(i) T is said to be monotone on C, if

$$\langle Tx - Ty, x - y \rangle \geq 0, \quad \text{for all } x, y \in C,$$

(ii) a mapping $T : C \to C$ is nonexpansive if

$$\|Tx - Ty\| \leq \|x - y\|, \quad \text{for all } x, y \in C.$$

We denote by $\mathrm{Fix}(T)$ the set of fixed points of T; that is, $\mathrm{Fix}(T) = \{x \in C : Tx = x\}$. Note that $\mathrm{Fix}(T)$ is always closed and convex (but maybe empty).

The metric projection from R^N onto C is the mapping $P_C : R^N \to C$, which assigns to each point $x \in C$ the unique point $P_C x \in C$ satisfying the property

$$\|x - P_C x\| = \inf_{y \in C} \|x - y\| =: d(x, C),$$

where $\| \cdot \|$ is the 2-norm.

The following properties of projections are useful and pertinent to our purpose.

Lemma 2.1 [13] *Given* $x \in R^N$

$$\text{(i)} \quad \langle x - P_C x, P_C x - y \rangle \geq 0, \quad \textit{for all } y \in C, \tag{2}$$

$$\text{(ii)} \quad \|x - P_C x\|^2 \leq \|x - y\|^2 - \|y - P_C x\|^2, \quad \textit{for all } y \in C, \tag{3}$$

$$\text{(iii)} \quad \langle P_C x - P_C y, x - y \rangle \geq \|P_C x - P_C y\|^2, \quad \textit{for all } y \in R^N. \tag{4}$$

Consequently, P_C is nonexpansive and monotone, and $I - P_C$ is also nonexpansive, then

$$\text{(iv)}\quad \big\langle (I-P_C)x-(I-P_C)y, x-y\big\rangle \geq \big\|(I-P_C)x-(I-P_C)y\big\|^2, \quad \textit{for all } y\in R^N. \tag{5}$$

Lemma 2.2 *For $\forall x, y \in R^N$*

$$\text{(i)}\quad \|x\pm y\|^2 = \|x\|^2 \pm 2\langle x,y\rangle + \|y\|^2, \tag{6}$$

$$\text{(ii)}\quad \big\|tx+(1-t)y\big\|^2 = t\|x\|^2 + (1-t)\|y\|^2 - t(1-t)\|x-y\|^2, \quad \forall t\in R^1. \tag{7}$$

Lemma 2.3 [14] *Let $U = I - \gamma A^T(I-P_Q)A$, where $\gamma \in (0, 2/L)$.*

(i) *U is an averaged operator; i.e. there exist some $\beta \in (0,1)$ and a nonexpansive operator V, $U = (1-\beta)U + \beta V$.*

(ii) $\mathrm{Fix}(U) = A^{-1}(Q)$, *then* $\mathrm{Fix}(P_C U) = C \cap A^{-1}(Q)$.

Proposition 2.1 [15] *For every $k \geq 0$, let $x^k \in R^N$, C_k and Q_k be defined as in* [7]. *Then for any $x\in R^N$ and $y \in R^M$ we have*

$$P_{C_k}(x) = \begin{cases} x - \frac{c(x^k)+\langle \xi^k, x-x^k\rangle}{\|\xi^k\|^2}\xi^k, & \textit{if } c(x^k)+\langle \xi^k, x-x^k\rangle > 0; \\ x, & \textit{otherwise} \end{cases}$$

and

$$P_{Q_k}(y) = \begin{cases} y - \frac{q(Ax^k)+\langle \eta^k, y-Ax^k\rangle}{\|\eta^k\|^2}\eta^k, & \textit{if } q(Ax^k)+\langle \eta^k, y-Ax^k\rangle > 0; \\ y, & \textit{otherwise.} \end{cases}$$

3 The preconditioning CQ algorithm

Stand [16] and Piana and Bertero [17] have applied the preconditioning matrix technologies to improve the Landweber and projected Landweber algorithms. The analyses deal with the operators A and A^*A, which are based on the singular value decomposition and a more general spectrum, respectively. We can also extend the technologies to improve the CQ algorithm.

As the SFP is to find a point $x^* \in C$, with $Ax^* \in Q$. Firstly, we set $\Omega = C \cap A^{-1}(Q)$, $A^{-1}(Q) = \{x^* \in R^N | Ax^* \in Q\}$. From Lemma 2.3, (1) can be depicted from the view of fixed point,

$$x^* = P_C\big(Ux^*\big). \tag{8}$$

Assume that $\Omega \neq \emptyset$, *i.e.* the SFP has a nonempty solution set, and x^* is the solution of SFP. Thus, we have

$$x^* = Ux^*,$$

so

$$A^T(I-P_Q)Ax^* = 0. \tag{9}$$

Then let $D : C \to C$ be a $N \times N$ symmetrical positive definite matrix, and $(AD)x^* \in Q$. Referring to (9), we can deduce that

$$\begin{aligned} DA^T(I - P_Q)ADx^* &= (AD)^T(I - P_Q)(AD)x^* \\ &= (AD)^T(AD)x^* - (AD)^T P_Q(AD)x^* \\ &= (AD)^T(AD)x^* - (AD)^T(AD)x^* = 0 \end{aligned} \tag{10}$$

or

$$x^* = x^* - \gamma DA^T(I - P_Q)ADx^* = U_D x^*.$$

Then U_D also has the same properties of U in Lemma 2.3, and we can obtain

$$x^* = P_C\left(U_D x^*\right). \tag{11}$$

Now we present a new algorithm, which is named a preconditioning CQ algorithm (PCQ).

Algorithm 3.1 Let $D : C \to C$ be a $N \times N$ symmetrical positive definite matrix, $x^0 \in C$ be arbitrary. For $k = 0, 1, \ldots$, calculate

$$x^{k+1} = P_C\left(x^k - \gamma DA^T(I - P_Q)ADx^k\right), \tag{12}$$

where $\gamma \in (0, 2/L)$, $L = \|DA^T\|^2$.

Algorithm 3.1 is to solve an extended SFP (ESFP), which can be represented as follows.

Definition 3.1 Let C and Q be nonempty closed convex sets in R^N and R^M, respectively, and A is an M by N matrix, D is an N by N symmetrical positive definite matrix, the ESFP is to find $x \in C$ with $ADx \in Q$. We denote the solution set of ESFP by G.

Remark 3.1 If we set $\tilde{x} = Dx \in D(C) = \tilde{C}$, then $A\tilde{x} \in Q$, the problem in Definition 3.1 is transformed into SFP.

Remark 3.2 If we set D is an unit matrix, then to find $x \in C$ with $Ax \in Q$, the problem in Definition 3.1 is transformed into SFP.

From Remark 3.1 we know that the SFP is to minimize the equation

$$f(\tilde{x}) = \frac{1}{2}\left\|(I - P_Q)A\tilde{x}\right\|^2, \quad \forall \tilde{x} \in \tilde{C}. \tag{13}$$

Substituting $\tilde{x} = Dx$ into (13), its gradient operator is

$$\nabla f(x) = DA^T(I - P_Q)ADx, \quad \forall x \in C.$$

While $C = R^N$ and $C \cap (AD)^{-1}(Q) \neq \emptyset$, we also have

$$\nabla f\left(x^*\right) = 0,$$

where $x^* \in R^N$ is the solution set of the extended SFP. We can obtain the following variational inequality:

$$\langle DA^T(I-P_Q)ADx^*, x-x^*\rangle \geq 0, \quad \forall x \in C.$$

Therefore, we have the next constrained least-squares problem:

$$\min\{f(x) : x \in C\}.$$

The following immediately follows.

Theorem 3.1 *Assume $G \neq \emptyset$, then $x^* \in G$, if and only if $x^* = \arg\min\{f(x)|x \in C\}$, if and only if $\langle \nabla f(x^*), x-x^*\rangle \geq 0$, $\forall x \in C$, where*

$$f(x) = \frac{1}{2}\|(I-P_Q)ADx\|^2, \quad \forall x \in C. \tag{14}$$

As $U_D = I - \gamma DA^T(I-P_Q)AD$, from (12) we have

$$x^0 \in C, \quad x^{k+1} = P_C(U_D x^k), \quad k = 0, 1, \ldots. \tag{15}$$

In order to establish the convergence of Algorithm 3.1, we need the following theorem.

Theorem 3.2 *Assume that $G = C \cap (AD)^{-1}(Q) \neq \emptyset$, then*
(i) $\mathrm{Fix}(U_D) = (AD)^{-1}(Q) = \{x \in R^N | ADx \in Q\}$;
(ii) $\mathrm{Fix}(P_C U_D) = G$.

Proof As $D^T = D$, $(AD)^T = DA^T$, we have $U_D = I - \gamma(AD)^T(I-P_Q)(AD)$.

Firstly, we prove $(AD)^{-1}(Q) \subset \mathrm{Fix}(U_D)$. For $\forall x \in (AD)^{-1}(Q)$, then $x \in R^N$ and $ADx \in Q$, we have $P_Q ADx = ADx$. So

$$U_D x = x - \gamma(AD)^T\big((AD)x - P_Q(AD)x\big) = x - 0 = x.$$

Therefore, $x \in \mathrm{Fix}(U_D)$.

Secondly, we prove $\mathrm{Fix}(U_D) \subset (AD)^{-1}(Q)$.

As $G = C \cap (AD)^{-1}(Q) \neq \emptyset$, we choose $z \in G$, then $z \in C$ and $z \in (AD)^{-1}(Q)$, so $(AD)z \in Q$.

For $\forall x \in \mathrm{Fix}(U_D)$, we have $A^T(I-P_Q)ADx = 0$. From the properties of a projection, we can deduce

$$\langle (I-P_Q)ADx, (AD)z - P_Q(AD)x\rangle \leq 0,$$

therefore,

$$\begin{aligned} \|(I-P_Q)ADx\|^2 &= \langle (I-P_Q)ADx, (I-P_Q)ADx\rangle \\ &= \langle (I-P_Q)ADx, (AD)x - (AD)z\rangle \\ &\quad + \langle (I-P_Q)ADx, (AD)z - P_Q(AD)x\rangle \end{aligned}$$

$$\begin{aligned}&\le \langle (I-P_Q)ADx, AD(x-z)\rangle \\ &= \langle A^T(I-P_Q)ADx, D(x-z)\rangle = 0,\end{aligned}$$

then $(I-P_Q)ADx = 0$, and $ADx = P_Q(AD)x \in Q$.

We obtain $x \in (AD)^{-1}(Q)$, thus (i) is proved.

We can also deduce that $\mathrm{Fix}(P_C U_D) = \mathrm{Fix}(P_C) \cap \mathrm{Fix}(U_D) \triangleq G = C \cap (AD)^{-1}(Q) \neq \emptyset$. □

Theorem 3.3 *Assume* $G \neq \emptyset$, $0 < \gamma < 2/L$, $L = \|DA^T\|^2$, *the sequence* $\{x^k\}$ *is generated by* (15), *there exists* $\lim_{k\to\infty} x^k \to x^* \in G$.

Proof Firstly, we show that if $\gamma = 2/L$, the operator

$$V = I - \frac{2}{L}DA^T(I-P_Q)AD \tag{16}$$

is nonexpansive.

For $\forall x, y \in C$, from (4) and (6) we have

$$\begin{aligned}\|Vx - Vy\|^2 &= \left\| x - y - \frac{2}{L}\bigl(DA^T(I-P_Q)ADx - DA^T(I-P_Q)ADy\bigr)\right\|^2 \\ &= \|x-y\|^2 - \frac{4}{L}\langle DA^T(I-P_Q)ADx - DA^T(I-P_Q)ADy, x-y\rangle \\ &\quad + \frac{4}{L^2}\left\|DA^T(I-P_Q)ADx - DA^T(I-P_Q)ADy\right\|^2 \\ &= \|x-y\|^2 - \frac{4}{L}\langle (I-P_Q)ADx - (I-P_Q)ADy, ADx - ADy\rangle \\ &\quad + \frac{4}{L^2}\left\|DA^T(I-P_Q)ADx - DA^T(I-P_Q)ADy\right\|^2 \\ &\le \|x-y\|^2 - \frac{4}{L}\left\|(I-P_Q)ADx - (I-P_Q)ADy\right\|^2 \\ &\quad + \frac{4}{L}\left\|(I-P_Q)ADx - (I-P_Q)ADy\right\|^2 \\ &= \|x-y\|^2,\end{aligned}$$

therefore,

$$\|Vx - Vy\|^2 \le \|x-y\|^2. \tag{17}$$

Next, we can easily obtain $0 < \gamma L/2 < 1$, and we set

$$\beta = \frac{\gamma L}{2} \in (0,1).$$

From (16) we deduce that

$$\begin{aligned}U_D &= I - \gamma DA^T(I-P_Q)AD \\ &= \left(1 - \frac{\gamma L}{2}\right)I + \frac{\gamma L}{2}\left(I - \frac{2}{L}DA^T(I-P_Q)AD\right) \\ &= (1-\beta)I + \beta V \qquad (18)\end{aligned}$$

and V is nonexpansive, hence, while $0 < \gamma < 2/L$, U_D is an averaged nonexpansive operator.

Finally, we choose $\forall p \in G$, where $p \in C$ and $p = U_D p$. We have $p = P_C U_D p = P_C p$. From (7), we have

$$\begin{aligned} \|x^{k+1} - p\|^2 &= \|P_C U_D x^k - P_C U_D p\|^2 \le \|U_D x^k - p\|^2 \\ &= \|(1-\beta)x^k + \beta V x^k - p\|^2 \\ &= \|(1-\beta)(x^k - p) + \beta(Vx^k - p)\|^2 \\ &= (1-\beta)\|x^k - p\|^2 + \beta\|Vx^k - p\|^2 - \beta(1-\beta)\|x^k - Vx^k\|^2 \\ &\le \|x^k - p\|^2 - \beta(1-\beta)\|x^k - Vx^k\|^2, \end{aligned} \tag{19}$$

which implies that $\{\|x^k - p\|^2\}$ is monotonically decreasing and hence $\lim_{k\to\infty} \|p - x^k\|^2 = d \ge 0$. Specially, $\{x^k\}$ is bounded.

From (19) we can deduce that

$$\beta(1-\beta)\|x^k - Vx^k\|^2 \le \|x^k - p\|^2 - \|x^{k+1} - p\|^2,$$

as $\|x^k - p\|^2 - \|x^{k+1} - p\|^2 \to 0$, we have $x^k - Vx^k \to 0$.

Let x^* be an arbitrary cluster point of the sequence $\{x^k\}$. Then there exists a subsequence $\{x^{k_j}\} \subset \{x^k\}$, then $x^{k_j} \to x^*$ $(j \to \infty)$. As $\{x^k\} \subset C$, $x^* \in C$, and $x^* = P_C x^*$. Because V is nonexpansive and continuous, then $Vx^{k_j} \to Vx^*$ $(j \to \infty)$.

As $\|x^* - Vx^*\| \le \|x^* - x^{k_j}\| + \|x^{k_j} - Vx^{k_j}\| + \|Vx^{k_j} - Vx^*\| \to 0$, we have $x^* = Vx^*$, then $U_D x^* = (1-\beta)x^* + \beta Vx^* = x^*$. Therefore, $x^* = P_C x^* = P_C U_D x^*$, from Theorem 3.2, we have $x^* \in G$. However, $\lim_{n\to\infty} \|x^k - x^*\| = d \ge 0$ exists, and there exists a subsequence $\{x^{k_j}\}$ of $\{x^k\}$ s.t. $x^{k_j} \to x^*$ $(j \to \infty)$, therefore, there must be $x^k \to x^*$ $(j \to \infty)$. □

4 Several extensions of the preconditioning CQ algorithm

In virtue of kinds of CQ-like algorithms for solving the SFP, we can also deduce the following meaningful results for solving the ESFP without proof.

According to the relaxed CQ algorithm [7], we firstly obtain the relaxed projection method.

Algorithm 4.1 Let $D : C_k \to C_k$ be a $N \times N$ symmetrical positive definite matrix, x^0 be arbitrary. For $k = 0, 1, \ldots$, calculate

$$x^{k+1} = P_{C_k}\bigl(x^k - \gamma DA^T(I - P_{Q_k})ADx^k\bigr), \tag{20}$$

where $\gamma \in (0, 2/L)$, $L = \|DA^T\|^2$.

Theorem 4.1 *Let $\{x^k\}$ be a sequence generated by the relaxed preconditioning CQ algorithm. Then $\{x^k\}$ converges to a solution of ESFP.*

Next, from the papers [8] and [14], define $\nabla f_k : R^N \to R^N$ by

$$\nabla f_k(x) = DA^T(I - P_{Q_k})ADx,$$

and we can obtain an adaptive algorithm with strong convergence.

Algorithm 4.2 Let $D : C_k \to C_k$ be a $N \times N$ symmetrical positive definite matrix, given constants $\lambda > 0$, $l \in (0,1)$, $\mu \in (0,1)$. Let x^0 be arbitrary, for $k = 0,1,\ldots$, let

$$\bar{x}^k = P_{C_k}\big(x^k - \rho_k DA^T(I - P_{Q_k})ADx^k\big), \tag{21}$$

where $\rho_k = \lambda l^{m_k}$ and m_k is the smallest nonnegative integer m such that

$$\rho_k \big\| \nabla f_k\big(x^k\big) - \nabla f\big(\bar{x}^k\big) \big\| \le \mu \big\| x^k - \bar{x}^k \big\|. \tag{22}$$

Set

$$x^{k+1} = P_{C_k}\big[(1 - \alpha_k)\big(x^k - \rho_k DA^T(I - P_{Q_k})AD\bar{x}^k\big)\big], \tag{23}$$

where $\{\alpha_k\}$ is a real sequence in $(0,1)$ that satisfies conditions (C_1) $\lim_{k\to\infty} \alpha_k = 0$; and (C_2) $\sum_{k=1}^{\infty} \alpha_k = \infty$.

Lemma 4.1 *For all* $k = 0,1,\ldots$, ∇f_k *is Lipschitz continuous on* R^N *with constant* L *and co-coercive on* R^N *with modulus* $1/L$, *where* L *is the largest eigenvalue of the matrix* A^TA. *Therefore*, *the Armijo-like search rule* (22) *is well defined*.

Lemma 4.2 *For all* $k = 0,1,\ldots$, $\frac{\mu l}{L} < \rho_k \le \gamma$.

Theorem 4.2 *Let* $\{x^k\}$ *be a sequence generated by Algorithm* 4.2. *If the solution set of the SFP is nonempty, then* $\{x^k\}$ *converges strongly to a solution of the ESFP.*

As there exists an Armijo-like search step in Algorithm 4.2, the complexity of the implementation will be increased. Next, we propose a new variable stepsize to improve Algorithm 3.1.

Algorithm 4.3 Let $D^k : C \to C$ be a variable $N \times N$ symmetrical positive definite matrix, $k = 0,1,\ldots$. For $\forall x^0 \in C$, calculate

$$x^{k+1} = P_C\big(x^k - \gamma^k D^k A^T(I - P_Q)AD^k x^k\big), \tag{24}$$

where $\gamma^k \in (0, 2/(L * M_D))$, L is the largest eigenvalue of A^TA, M_D is the minimum value of all the largest eigenvalues of D^k, for $k = 0,1,\ldots$. Specially, set $\gamma^k = 1/(L * M_D)$.

5 Approximating a variable preconditioner

In the above algorithms, the preconditioner D is continuous, positive definite and bounded so that it has a continuous inverse [17]. According to the preconditioning CQ algorithm, we set D commutes with the operator A^TA. Therefore, we set a matrix function F with positive value, and its dimension should be consistent with $\dim(A^TA)$. Moreover, F should have a positive lower bound in order to satisfy the existing inverse. Strand [16] also assumes that F is a polynomial or a rational function. Thus, the operator D is given by

$$D = F\big(A^TA\big). \tag{25}$$

Assume $(A^TA)^{-1}$ is existed, the product DA^TAD is without restrictions in (10), we can deduced that sometimes D should be chose close to $(A^TA)^{-1/2}$ as more as possible.

The best condition is to calculate $(A^TA)^{-1}$ exactly, but it is always hard with the signal and image reconstruction problems. If F be a polynomial function, Stand [16] have provided an example with seventh-order, and Neumann's series approach can also express $(A^TA)^{-1}$. However, the polynomial method needs to calculate the high order matrix multiplication, therefore, it can not be implemented easily.

If we choose F be a rational function, a simple example, closely related to the Tikhonov regularization method has been used in [17]. According to the example, the approximate inverse preconditioner D can be given by

$$D = \mathrm{Re}\left[\left(A^TA + \alpha I\right)^{-1/2}\right], \tag{26}$$

where Re denotes the real part, α is a positive real parameter, and good choices of α may be much smaller than the values provided by the methods used for estimating optimum values of the Tikhonov regularization parameter.

As (26) involves the matrix inverse, we next propose a diagonal format of D that does not calculate matrix inverses. Furthermore, the choice of D is related to the convergence properties of the algorithm. If D is evolutive following the iterations, the convergence rate of algorithm will also be accelerated.

From (10), we can deduce that

$$A^TA\tilde{x} = A^TP_{Q_k}(A\tilde{x}), \quad \tilde{x} \in \Omega. \tag{27}$$

As D should be chosen closely to $(A^TA)^{-1/2}$, we assume λ is the approximate eigenvalue matrix of A^TA, and then we set

$$\lambda_{j\times j}\tilde{x} \triangleq A^TP_{Q_k}(A\tilde{x}), \quad j = 1, 2, \ldots, N. \tag{28}$$

Therefore, we can obtain the approximate variable preconditioners with respect to $(A^TA)^{-1}$ on the $(k+1)$th iteration:

$$\bar{D}_{jj}^{k+1} = \begin{cases} x_j^k/(A^TP_{Q_k}Ax^k)_j, & \text{if } x_j^k \neq 0 \text{ and } (A^TP_{Q_k}Ax^k)_j \neq 0; \\ \bar{D}_{jj}^k & \text{otherwise.} \end{cases} \tag{29}$$

Then we can also get the approximate variable preconditioners with respect to $(A^TA)^{-1/2}$ on the $(k+1)$th iteration:

$$D_{jj}^{k+1} = \mathrm{Re}\left[\left(\bar{D}_{jj}^{k+1}\right)^{1/2}\right]. \tag{30}$$

Otherwise, a variable stepsize in Algorithm 4.3 can be estimated. Set L_{D^k} is the largest eigenvalue of D^k, $k = 0, 1, \ldots$, on the kth iteration, we have $M_{D^k} = \min\{L_{D^n} | n = 0, 1, \ldots, k\}$. Then set $l_{\bar{D}^k}$, the minimum eigenvalue of $\bar{D}^k$, $k = 0, 1, \ldots$, on the kth iteration, we have $L_k = \max\{1/l_{\bar{D}^n} | n = 0, 1, \ldots, k\}$. Therefore, a variable stepsize with respect to Algorithm 4.3 can be approximated by

$$\gamma_k = 1/(L_k * M_{D^k}). \tag{31}$$

6 Numerical results

In this section, we present some numerical results for the proposed method. The following three examples are taken from the test problems in [15]. For Examples 6.2 and 6.3, we should first transform into the ESFP. The stopping criterion is $\|x^{k+1} - x^k\| < \varepsilon$, and we took $\varepsilon = 10^{-10}$, $\gamma = 1/\|DA^T\|^2$, $\alpha = 0.1$. The projections are computed by Proposition 2.1.

Algorithm 4.1 was implemented in the Matlab R2011b (Windows version) programming environment. The codes were ran on a PC with 1.98 GB memory and Intel(R) Pentium(R) dual-core CPU G630 running at 2.69 GHz. The iteration numbers and the computational time for the methods in Section 5 with different starting points are given in Tables 1, 2, and 3, all CPU times reported are in seconds.

Example 6.1 (A convex feasibility problem, CFP) Let $C = \{x \in R^3 | x_2^2 - x_3^2 - 4 \leq 0\}$, $Q = \{y \in R^3 | y_3 - 1 - y_1^2 \leq 0\}$. Find some point x in $C \cap Q$.

Example 6.2 (A split feasibility problem) Let $A = \begin{pmatrix} 2 & -1 & 3 \\ 4 & 2 & 5 \\ 2 & 0 & 2 \end{pmatrix}$, $C = \{x \in R^3 | x_1 + x_2^2 + 2x_3 \leq 0\}$, $Q = \{y \in R^3 | y_1^2 + y_2 - y_3 \leq 0\}$. Find $x \in C$ with $Ax \in Q$.

Example 6.3 (A split feasibility problem) Let $A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 2 & 0 & 1 \end{pmatrix}$, $C = \{x \in R^3 | x_1 + x_2^2 + 2x_3 \leq 0\}$, $Q = \{y \in R^3 | y_1^2 + y_2 - y_3 \leq 0\}$. Find $x \in C$ with $Ax \in Q$.

Compared with the results in [15], we can obtain:

(1) For the CFP, as $A = I$, the relaxed preconditioning method played a role inconspicuously.

(2) For the SFP, the results are better than the ones in [15], and when A is not sparse the effect is obvious.

Table 1 Numerical results for Example 6.1

Starting point	The method (26)			The method (30)		
	k	**CPU (s)**	**Approximate solution**	***k***	**CPU (s)**	**Approximate solution**
(1,2,3)′	6	0.0156	(1.4000, 1.1864, 1.6101)′	6	0.0262	(1.1798, 1.1380, 1.6447)′
(1,1,1)′	1	0.0149	(1,1,1)′	1	0.0252	(1,1,1)′
rand(3,1)′ ∗ 10	7	0.0153	(9.5717, 1.0372, 1.7101)′	7	0.0270	(9.5717, 1.0372, 1.7101)′

Table 2 Numerical results for Example 6.2

Starting point	The method (26)			The method (30)		
	k	**CPU (s)**	**Approximate solution**	***k***	**CPU (s)**	**Approximate solution**
(1,2,3)′	19	0.0166	(−0.3097, −0.1638, 0.1067)′	5	0.0264	(−0.6409, −0.0027, 0.3205)′
(1,1,1)′	5	0.0155	(0.3988, 0.0763, −0.2023)′	10	0.0271	(0.1828, 0.0493, −0.1644)′
rand(3,1)′ ∗ 10	7	0.0159	(6.0095, −0.0363, −3.0054)′	7	0.0280	(3.7284, 0.4046, −2.6730)′

Table 3 Numerical results for Example 6.3

Starting point	The method (26)			The method (30)		
	k	**CPU (s)**	**Approximate solution**	***k***	**CPU (s)**	**Approximate solution**
(1,2,3)′	6	0.0156	(1.4000, 1.1864, 1.6101)′	8	0.0274	(−0.0500, 0.0480, 0.0234)′
(1,1,1)′	1	0.0149	(1,1,1)′	4	0.0269	(0.3412, 0.0403, −0.1714)′
rand(3,1)′ ∗ 10	7	0.0153	(9.5717, 1.0372, 1.7101)′	7	0.0273	(0.9290, −0.2692, −0.5000)′

7 Conclusions

In this paper, by adopting the preconditioning techniques, a modified CQ algorithm is named preconditioning CQ algorithm, and its extensions for solving the ESFP have been presented. The approximate methods for how to estimate the preconditioner D are also discussed; the approximate diagonal preconditioner method does not need to compute the matrix inverses and the largest eigenvalue of the matrix A^TA. Thus, the algorithm can be implemented easily. Moreover, the corresponding convergence property has been established in the feasible case of ESFP. The numerical results showed that the proposed algorithms and methods are effective to solve some problems.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
Both authors take equal roles in deriving results and writing of this paper. Both authors read and approved the final manuscript.

Author details
[1]Department of Mathematics, Shijiazhuang Mechanical Engineering College, Shijiazhuang, 050003, China. [2]Department of Mathematics and Information, Hebei Normal University, Shijiazhuang, 050016, China.

Acknowledgements
The authors would like to thank the associate editor and the referees for their comments and suggestions. This research was supported by the National Natural Science Foundation of China (11071053).

Received: 27 October 2013 Accepted: 3 April 2014 Published: 06 May 2014

References

1. Censor, Y, Elfving, T: A multiprojection algorithm using Bregman projections in a product space. Numer. Algorithms **8**, 221-239 (1994)
2. Bauschke, HH, Borwein, JM: On projection algorithms for solving convex feasibility problems. SIAM Rev. **38**, 367-426 (1996)
3. Byrne, CL: A unified treatment of some iterative algorithms in signal processing and image reconstruction. Inverse Probl. **20**, 103-120 (2004)
4. Byrne, CL: Iterative projection onto convex sets using multiple Bregman distances. Inverse Probl. **15**, 1295-1313 (1999)
5. Byrne, CL: Bregman-Legendre multidistances projection algorithm for convex feasibility and optimization. In: Butnariu, D, Censor, Y, Reich, S (eds.) Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications, pp. 87-100. Elsevier, Amsterdam (2001)
6. Byrne, CL: Iterative oblique projection onto convex sets and the split feasibility problem. Inverse Probl. **18**, 441-453 (2002)
7. Yang, QZ: The relaxed CQ algorithm solving the split feasibility problem. Inverse Probl. **20**, 1261-1266 (2004)
8. Qu, B, Xiu, N: A note on the CQ algorithm for the split feasibility problem. Inverse Probl. **21**, 1655-1665 (2005)
9. Wang, Z, Yang, Q: The relaxed inexact projection methods for the split feasibility problem. Appl. Math. Comput. **217**, 5347-5359 (2011)
10. Yang, Q, Zhao, J: The projection-type methods for solving the split feasibility problem. Math. Numer. Sin. **28**, 121-132 (2006) (in Chinese)
11. Xu, HK: Iterative methods for the split feasibility problem in infinite-dimensional Hilbert spaces. Inverse Probl. **26**, 1-17 (2010)
12. López, G, Martín-Márquez, V, Wang, F, Xu, H-K: Solving the split feasibility problem without prior knowledge of matrix norms. Inverse Probl. **28**, 1-18 (2012)
13. Zarantonello, EH: Projections on convex sets in Hilbert space and spectral theory. In: Zarantonello, EH (ed.) Contributions to Nonlinear Functional Analysis. Academic Press, New York (1971)
14. Wang, F, Xu, HK: Approximating curve and strong convergence of the CQ algorithm for the split feasibility problem. J. Inequal. Appl. **2010**, Article ID 102085 (2010)
15. Abdellah, B, Muhammad, AN: On descent-projection method for solving the split feasibility problems. J. Glob. Optim. **54**, 627-639 (2012)
16. Strand, ON: Theory and methods related to the singular-function expansion and Landweber's iteration for integral equations of the first kind. SIAM J. Numer. Anal. **11**, 798-825 (1974)
17. Piana, M, Bertero, M: Projected Landweber method and preconditioning. Inverse Probl. **13**, 441-463 (1997)

10.1186/1029-242X-2014-163
Cite this article as: Wang and Zhou: A preconditioning method of the CQ algorithm for solving an extended split feasibility problem. *Journal of Inequalities and Applications* 2014, 2014:163