*mathematics*

MDPI

*Article*

# A New Hybrid CQ Algorithm for the Split Feasibility Problem in Hilbert Spaces and Its Applications to Compressed Sensing

**Suthep Suantai [1], Suparat Kesornprom [2,*] and Prasit Cholamjiak [2,*]**

[1] Research Center in Mathematics and Applied Mathematics, Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand; suthep.s@cmu.ac.th

[2] School of Science, University of Phayao, Phayao 56000, Thailand

* Correspondence: suparat.ke@gmail.com (S.K.); prasit.ch@up.ac.th (P.C.)

Received: 24 June 2019; Accepted: 23 August 2019; Published: 27 August 2019

check for updates

**Abstract:** In this paper, we focus on studying the split feasibility problem (SFP), which has many applications in signal processing and image reconstruction. A popular technique is to employ the iterative method which is so called the relaxed CQ algorithm. However, the speed of convergence usually depends on the way of selecting the step size of such algorithms. We aim to suggest a new hybrid CQ algorithm for the SFP by using the self adaptive and the line-search techniques. There is no computation on the inverse and the spectral radius of a matrix. We then prove the weak convergence theorem under mild conditions. Numerical experiments are included to illustrate its performance in compressed sensing. Some comparisons are also given to show the efficiency with other CQ methods in the literature.

**Keywords:** split feasibility problem; CQ algorithm; gradient method; line-search

## 1. Introduction

In the present work, we aim to study the split feasibility problem (SFP), which is to find a point

$$x^* \in C \text{ such that } Ax^* \in Q \tag{1}$$

where $C$ and $Q$ are nonempty closed and convex subsets of real Hilbert spaces $H_1$ and $H_2$, respectively, and $A : H_1 \to H_2$ a bounded linear operator. In 1994, the SFP was first investigated by Censor and Elfving [1] in finite dimensional Hilbert spaces. There have been applications in real world such as image processing and signal recovery (see [2,3]). Byrne [4,5] introduced the following recursive procedure for solving SFP:

$$x_{n+1} = P_C(x_n - \alpha_n A^*(I - P_Q)Ax_n) \tag{2}$$

where $\{\alpha_n\} \subset (0, 2/\|A\|^2)$, $P_C$ and $P_Q$ are the projections onto $C$ and $Q$, respectively, and $A^*$ is the adjoint of $A$. This projection algorithm is usually called the CQ algorithm. Subsequently, Yang [6] introduced the relaxed CQ algorithm. In this case, the projections $P_C$ and $P_Q$ are, respectively, replaced by $P_{C_n}$ and $P_{Q_n}$, where

$$C_n = \{x \in H_1 : c(x_n) + \langle \xi_n, x - x_n \rangle \leq 0\}, \tag{3}$$

where $c : H_1 \to \mathbb{R}$ is convex and lower semicontinuous, and $\xi_n \in \partial c(x_n)$, and

$$Q_n = \{y \in H_2 : q(Ax_n) + \langle \eta_n, y - Ax_n \rangle \leq 0\}, \tag{4}$$

where $q : H_2 \to \mathbb{R}$ is convex and lower semicontinuous, and $\eta_n \in \partial q(Ax_n)$. In what follows, we define

$$f_n(x) = \frac{1}{2}\|(I - P_{Q_n})Ax\|^2,\ n \geq 1 \tag{5}$$

and

$$\nabla f_n(x) = A^*(I - P_{Q_n})Ax. \tag{6}$$

Precisely, Yang [6] proposed the relaxed CQ algorithm in a finite-dimensional Hilbert space as follows:

**Algorithm 1.** *Let $x_1 \in H_1$. For $n \geq 1$, define*

$$x_{n+1} = P_{C_n}(x_n - \alpha_n \nabla f_n(x_n)). \tag{7}$$

*where $\{\alpha_n\} \subset (0, 2/\|A\|^2)$.*

It is seen that, since the sets $C_n$ and $Q_n$ are half spaces, the projections are easily to be computed. However, the step size $\{\alpha_n\}$ still depends on the norm of $A$.

To eliminate this difficulty, in 2012, López et al. [7] suggested a new way to select the step size $\alpha_n$ as follows:

$$\alpha_n = \frac{\beta_n f_n(x_n)}{\|\nabla f_n(x_n)\|^2} \tag{8}$$

where $\{\beta_n\}$ is a sequence in $(0, 4)$ such that $0 < a \leq \liminf_{n\to\infty} \beta_n \leq \limsup_{n\to\infty} \beta_n \leq b < 4$ for some $a, b \in (0, 4)$. They established the weak convergence of the CQ algorithm (Equation (2)) and the relaxed CQ algorithm (Equation (7)) with the step size defined by Equation (8) in real Hilbert spaces.

Qu and Xiu [8] adopted the line-search technique to construct the step size in Euclidean spaces as follows:

**Algorithm 2.** *Choose $\sigma > 0$, $\rho \in (0, 1)$, $\mu \in (0, 1)$. Let $x_1$ be a point in $H_1$. For $n \geq 1$, let*

$$y_n = P_{C_n}(x_n - \alpha_n \nabla f_n(x_n)), \tag{9}$$

*where $\alpha_n = \sigma\rho^{m_n}$ and $m_n$ is the smallest nonnegative integer such that*

$$\alpha_n\|\nabla f_n(x_n) - \nabla f_n(y_n)\| \leq \mu\|x_n - y_n\|. \tag{10}$$

*Set*

$$x_{n+1} = P_{C_n}(x_n - \alpha_n \nabla f_n(y_n)). \tag{11}$$

In 2012, Bnouhachem et al. [9] proposed the following projection method for solving the SFP.

**Algorithm 3.** *For a given $x_1 \in \mathbb{R}^n$, let*

$$y_n = P_{C_n}(x_n - \alpha_n \nabla f_n(x_n)) \tag{12}$$

*where $\alpha_n > 0$ satisfies*

$$\alpha_n\|\nabla f_n(x_n) - \nabla f_n(y_n)\| \leq \mu\|x_n - y_n\|,\ 0 < \mu < 1. \tag{13}$$

*Define*

$$x_{n+1} = P_{C_n}(x_n - \varphi_n d(x_n, \alpha_n)) \tag{14}$$

*where*

$$\begin{aligned} d(x_n, \alpha_n) &= x_n - y_n + \alpha_n \nabla f_n(y_n) \\ \varepsilon_n &= \alpha_n(\nabla f_n(y_n) - \nabla f_n(x_n)) \\ D(x_n, \alpha_n) &= x_n - y_n - \varepsilon_n \\ \phi(x_n, \alpha_n) &= \langle x_n - y_n, D(x_n, \alpha_n) \rangle \end{aligned} \tag{15}$$

*and*

$$\varphi_n = \frac{\phi(x_n, \alpha_n)}{\|d(x_n, \alpha_n)\|^2}. \tag{16}$$

Recently, many authors establish weak and strong convergence theorems for the SFP (see also [10,11]).

In this work, combining the work of Bnouhachem et al. [9] and López et al. [7], we suggest a new hybrid CQ algorithm for solving the split feasibility problem and establish weak convergence theorem in Hilbert spaces. Finally, numerical results are given for supporting our main results. The comparison is also given to algorithms of Qu and Xiu [8] and Bnouhachem et al. [9]. It is shown that our method has a better convergence behavior than these CQ algorithms through numerical examples.

## 2. Preliminaries

We next recall some useful basic concepts that will be used in our proof. Let $H$ be a real Hilbert space equipped with the inner product $\langle \cdot, \cdot \rangle$ and the norm $\| \cdot \|$. Let $T : H \to H$ be a nonlinear mapping. Then, $T$ is called *firmly nonexpansive* if, for all $x, y \in H$,

$$\langle x - y, Tx - Ty \rangle \geq \|Tx - Ty\|^2. \tag{17}$$

In a real Hilbert space $H$, we have the following equality:

$$\langle x, y \rangle = \frac{1}{2}\|x\|^2 + \frac{1}{2}\|y\|^2 - \frac{1}{2}\|x - y\|^2. \tag{18}$$

A function $f : H \to \mathbb{R}$ is convex if and only if

$$f(z) \geq f(x) + \langle \nabla f(x), z - x \rangle \tag{19}$$

for all $z \in H$.

A function $f : H \to \mathbb{R}$ is said to be *weakly lower semi-continuous* (w-lsc) at $x$ if $x_n \rightharpoonup x$ implies

$$f(x) \leq \liminf_{n \to \infty} f(x_n). \tag{20}$$

The projection of a nonempty, closed and convex set $C$ onto $H$ is defined by

$$P_C x := \arg\min_{y \in C} \|x - y\|^2, \ x \in H. \tag{21}$$

We note that $P_C$ and $I - P_C$ are firmly nonexpansive. From [5], we know that, if

$$f(x) = \frac{1}{2}\|(I - P_Q)Ax\|^2,$$

then $\nabla f$ is $\|A\|^2$-Lipschitz continuous. Moreover, in real Hilbert spaces, we know that [12]

(i) $\langle x - P_C x, z - P_C x \rangle \leq 0$ for all $z \in C$;

(ii) $\|P_C x - P_C y\|^2 \leq \langle P_C x - P_C y, x - y \rangle$ for all $x, y \in H$; and

(iii) $\|P_C x - z\|^2 \leq \|x - z\|^2 - \|P_C x - x\|^2$ for all $z \in C$.

**Lemma 1.** *[12] Let $S$ be a nonempty, closed and convex subset of a real Hilbert space $H$ and $\{x_n\}$ be a sequence in $H$ that satisfies the following assumptions:*

*(i) $\lim_{n\to\infty} \|x_n - x\|$ exists for each $x \in S$; and*

*(ii) $\omega_w(x_n) \subset S$.*

*Then, $\{x_n\}$ converges weakly to a point in $S$.*

## 3. Main Results

Throughout this paper, let $S$ be the set of solution of SFP and suppose that $S$ is nonempty. Let $C$ and $Q$ be nonempty that satisfy the following assumptions:

(A1) The set $C$ is defined by

$$C = \{x \in H_1 : c(x) \leq 0\}, \tag{22}$$

where $c : H_1 \to \mathbb{R}$ is convex, subdifferentiable on $C$ and bounded on bounded sets, and the set $Q$ is defined by

$$Q = \{y \in H_2 : q(y) \leq 0\}, \tag{23}$$

where $q : H_2 \to \mathbb{R}$ is convex, subdifferentiable on $Q$ and bounded on bounded sets.

(A2) For each $x \in H_1$ , at least one subgradient $\xi \in \partial c(x)$ can be computed, where

$$\partial c(x) = \{z \in H_1 : c(u) \geq c(x) + \langle u - x, z\rangle, \forall u \in H_1\}. \tag{24}$$

(A3) For each $y \in H_2$, at least one subgradient $\eta \in \partial q(y)$ can be computed, where

$$\partial q(x) = \{w \in H_2 : q(u) \geq q(y) + \langle v - y, w\rangle, \forall v \in H_2\}. \tag{25}$$

Next, we propose our new relaxed CQ algorithm in real Hilbert spaces.

**Algorithm 4.** *Let $x_1 \in H_1$, for any $\sigma > 0$, $\rho \in (0, 1)$, $\mu \in (0, \frac{1}{2})$. Assume $\{x_n\}$ and $\{y_n\}$ have been constructed. Compute $x_{n+1}$ via the formula*

$$y_n = P_{C_n}(x_n - \alpha_n \nabla f_n(x_n)), \tag{26}$$

*where $\alpha_n = \sigma\rho^{m_n}$ and $m_n$ is the smallest nonnegative integer such that*

$$\alpha_n \|\nabla f_n(x_n) - \nabla f_n(y_n)\| \leq \mu \|x_n - y_n\|. \tag{27}$$

*Define*

$$x_{n+1} = y_n - \tau_n \nabla f_n(y_n) \tag{28}$$

*where*

$$\tau_n = \frac{\beta_n f_n(y_n)}{\|\nabla f_n(y_n)\|^2 + \theta_n},\ 0 < \beta_n < 4,\ 0 < \theta_n < 1. \tag{29}$$

**Lemma 2.** *[8] The line-search in Equation (27) terminates after a finite number of steps. In addition, we have the following:*

$$\frac{\mu\rho}{L} < \alpha_n \leq \sigma \tag{30}$$

*for all $n \geq 1$, where $L = \|A\|^2$.*

Next, we state our main theorem in this paper.

**Theorem 1.** *Assume that $\{\theta_n\}$ and $\{\beta_n\}$ satisfy the assumptions:*

*(a1)* $\lim_{n\to\infty} \theta_n = 0$; *and*

*(a2)* $\liminf_{n\to\infty} \beta_n(4-\beta_n) > 0$.

*Then, $\{x_n\}$ defined by Algorithm 4 converges weakly to a solution of the SFP.*

**Proof.** Let $z \in S$. Then, we have $z = P_{C_n}(z)$ and $Az = P_{Q_n}(Az)$. It follows that $\nabla f_n(z) = 0$. We see that

$$\begin{aligned} \|x_{n+1} - z\|^2 &= \|y_n - \tau_n \nabla f_n(y_n) - z\|^2 \\ &= \|y_n - z\|^2 + \tau_n^2 \|\nabla f_n(y_n)\|^2 - 2\tau_n \langle y_n - z, \nabla f_n(y_n)\rangle. \end{aligned} \tag{31}$$

Since $I - P_{Q_n}$ is firmly nonexpansive and $\nabla f_n(z) = 0$, we get

$$\begin{aligned} \langle y_n - z, \nabla f_n(y_n)\rangle &= \langle y_n - z, \nabla f_n(y_n) - \nabla f_n(z)\rangle \\ &= \langle y_n - z, A^*(I - P_{Q_n})Ay_n - A^*(I - P_{Q_n})Az\rangle \\ &= \langle Ay_n - Az, (I - P_{Q_n})Ay_n - (I - P_{Q_n})Az\rangle \\ &\geq \|(I - P_{Q_n})Ay_n\|^2 \\ &= 2f_n(y_n). \end{aligned} \tag{32}$$

It also follows that

$$\langle x_n - z, \nabla f_n(x_n)\rangle \geq 2f_n(x_n). \tag{33}$$

From Equation (19), we see that

$$\begin{aligned} 2\alpha_n \langle y_n - x_n, \nabla f_n(x_n)\rangle &= 2\alpha_n \langle y_n - x_n, \nabla f_n(x_n) - \nabla f_n(y_n)\rangle + 2\alpha_n \langle y_n - x_n, \nabla f_n(y_n)\rangle \\ &\geq -2\alpha_n \|y_n - x_n\| \|\nabla f_n(x_n) - \nabla f_n(y_n)\| \\ &\quad + 2\alpha_n \frac{1}{2}(\|(I - P_{Q_n})Ay_n\|^2 - \|(I - P_{Q_n})Ax_n\|^2) \\ &\geq -2\alpha_n \|y_n - x_n\| \|\nabla f_n(x_n) - \nabla f_n(y_n)\| - 2\alpha_n f_n(x_n). \end{aligned} \tag{34}$$

From Equations (33) and (34), we obtain

$$\begin{aligned} \|y_n - z\|^2 &= \|P_{C_n}(x_n - \alpha_n \nabla f_n(x_n)) - z\|^2 \\ &\leq \|x_n - \alpha_n \nabla f_n(x_n) - z\|^2 - \|y_n - x_n + \alpha_n \nabla f_n(x_n)\|^2 \\ &= \|x_n - z\|^2 + \|\alpha_n \nabla f_n(x_n)\|^2 - 2\alpha_n \langle x_n - z, \nabla f_n(x_n)\rangle \\ &\quad - \|y_n - x_n\|^2 - \|\alpha_n \nabla f_n(x_n)\|^2 - 2\alpha_n \langle y_n - x_n, \nabla f_n(x_n)\rangle \\ &= \|x_n - z\|^2 - 2\alpha_n \langle x_n - z, \nabla f_n(x_n)\rangle - \|y_n - x_n\|^2 - 2\alpha_n \langle y_n - x_n, \nabla f_n(x_n)\rangle \\ &\leq \|x_n - z\|^2 - 4\alpha_n f_n(x_n) - \|y_n - x_n\|^2 \\ &\quad + 2\alpha_n \|y_n - x_n\| \|\nabla f_n(x_n) - \nabla f_n(y_n)\| + 2\alpha_n f_n(x_n) \\ &\leq \|x_n - z\|^2 - 2\alpha_n f_n(x_n) - \|y_n - x_n\|^2 + 2\mu \|y_n - x_n\|^2 \\ &= \|x_n - z\|^2 - 2\alpha_n f_n(x_n) - (1 - 2\mu)\|y_n - x_n\|^2. \end{aligned} \tag{35}$$

Combining Equations (31), (32) and (35), we get

$$\begin{aligned}
\|x_{n+1}-z\|^2 &\leq \|x_n-z\|^2 - 2\alpha_n f_n(x_n) - (1-2\mu)\|y_n-x_n\|^2 + \tau_n^2\|\nabla f_n(y_n)\|^2 - 4\tau_n f_n(y_n) \\
&= \|x_n-z\|^2 - 2\alpha_n f_n(x_n) - (1-2\mu)\|y_n-x_n\|^2 \\
&\quad + \frac{\beta_n^2 f_n^2(y_n)}{(\|\nabla f_n(y_n)\|^2+\theta_n)^2}\|\nabla f_n(y_n)\|^2 - \frac{4\beta_n f_n^2(y_n)}{\|\nabla f_n(y_n)\|^2+\theta_n} \\
&\leq \|x_n-z\|^2 - 2\alpha_n f_n(x_n) - (1-2\mu)\|y_n-x_n\|^2 \\
&\quad + \frac{\beta_n^2 f_n^2(y_n)}{\|\nabla f_n(y_n)\|^2+\theta_n} - \frac{4\beta_n f_n^2(y_n)}{\|\nabla f_n(y_n)\|^2+\theta_n} \\
&= \|x_n-z\|^2 - 2\alpha_n f_n(x_n) - (1-2\mu)\|y_n-x_n\|^2 \\
&\quad -\beta_n(4-\beta_n)\frac{f_n^2(y_n)}{\|\nabla f_n(y_n)\|^2+\theta_n} \\
&\leq \|x_n-z\|^2 - 2\frac{\mu\ell}{L} f_n(x_n) - (1-2\mu)\|y_n-x_n\|^2 \\
&\quad -\beta_n(4-\beta_n)\frac{f_n^2(y_n)}{\|\nabla f_n(y_n)\|^2+\theta_n}, \qquad (36)
\end{aligned}$$

where the last inequality follows from Lemma 2. Since $0<\beta_n<4$ and $0<\mu<\frac{1}{2}$, it follows that

$$\|x_{n+1}-z\| \leq \|x_n-z\|. \qquad (37)$$

Thus, $\lim_{n\to\infty}\|x_n-z\|$ exists and hence $\{x_n\}$ is bounded.
From Equation (36) and Assumption (A2), it also follows that

$$\lim_{n\to\infty}\frac{f_n^2(y_n)}{\|\nabla f_n(y_n)\|^2+\theta_n}=0. \qquad (38)$$

By Assumption (A1), we have

$$\lim_{n\to\infty}\frac{f_n^2(y_n)}{\|\nabla f_n(y_n)\|^2}=0. \qquad (39)$$

It follows that

$$\lim_{n\to\infty} f_n(y_n) = \lim_{n\to\infty}\|(I-P_{Q_n})Ay_n\| = 0, \qquad (40)$$

and

$$\lim_{n\to\infty} f_n(x_n) = \lim_{n\to\infty}\|(I-P_{Q_n})Ax_n\| = 0. \qquad (41)$$

From Equation (36), we have

$$\lim_{n\to\infty}\|y_n-x_n\| = 0. \qquad (42)$$

Using Equations (40) and (42), we have

$$\begin{aligned}
\|Ax_n - P_{Q_n}Ay_n\| &= \|Ax_n - Ay_n + Ay_n - P_{Q_n}Ay_n\| \\
&\leq \|Ax_n - Ay_n\| + \|Ay_n - P_{Q_n}Ay_n\| \\
&= \|A\|\|x_n-y_n\| + \|Ay_n - P_{Q_n}Ay_n\| \\
&\to 0 \text{ as } n\to\infty. \qquad (43)
\end{aligned}$$

Let $x^*$ be a cluster point of $\{x_n\}$ with $\{x_{n_k}\}$ converging to $x^*$. From Equation (42), we see that $\{y_{n_k}\}$ also converges to $x^*$. We next show that $x^*$ is in $S$. Since $y_{n_k} \in C_{n_k}$, by the definition of $C_{n_k}$, we have

$$c(x_{n_k}) + \langle \xi_{n_k}, y_{n_k} - x_{n_k} \rangle \leq 0 \tag{44}$$

where $\xi_{n_k} \in \partial c(x_{n_k})$. By the assumption that $\{\xi_{n_k}\}$ is bounded and Equation (42), we get

$$\begin{aligned} c(x_{n_k}) &\leq \langle \xi_{n_k}, x_{n_k} - y_{n_k} \rangle \\ &\leq \|\xi_{n_k}\| \|x_{n_k} - y_{n_k}\| \\ &\to 0 \text{ as } k \to \infty \end{aligned} \tag{45}$$

which implies $c(x^*) \leq 0$. Hence $x^* \in C$. Since $P_{Q_{n_k}}(Ay_{n_k}) \in Q_{n_k}$, we obtain

$$q(Ax_{n_k}) + \langle \eta_{n_k}, P_{Q_{n_k}} Ay_{n_k} - Ax_{n_k} \rangle \leq 0 \tag{46}$$

where $\eta_{n_k} \in \partial q(Ax_{n_k})$. By the boundedness of $\{\eta_{n_k}\}$ and Equation (43), it follows that

$$\begin{aligned} q(Ax_{n_k}) &\leq \langle \eta_{n_k}, Ax_{n_k} - P_{Q_{n_k}} Ay_{n_k} \rangle \\ &\leq \|\eta_{n_k}\| \|Ax_{n_k} - P_{Q_{n_k}} Ay_{n_k}\| \\ &\to 0 \text{ as } k \to \infty. \end{aligned} \tag{47}$$

We conclude that $q(Ax^*) \leq 0$. Thus, $Ax^* \in Q$. Thus, $x^*$ is a solution of the SFP. Hence, by Lemma 1, we conclude that the sequence $\{x_n\}$ converges to a point in $S$. This completes the proof. □

## 4. Numerical Experiments

In this section, we provide numerical experiments in compressed sensing. We illustrate the performance of Algorithms 4 and 1 of Yang [6], Algorithm 2 of Qu and Xiu [8], and Algorithm 3 of Bnouhuchem et al. [9]. In signal processing, compressed sensing can be modeled as the following linear equation:

$$y = Ax + \varepsilon, \tag{48}$$

where $x \in \mathbb{R}^N$ is a recovered vector with $m$ nonzero components, $y \in \mathbb{R}^M$ is the observed data, $\varepsilon$ is the noisy and $A$ is an $M \times N$ matrix with $M < N$. The problem in Equation (48) can be seen as the LASSO problem:

$$\min_{x \in \mathbb{R}^N} \frac{1}{2} \|y - Ax\|^2 \text{ subject to } \|x\|_1 \leq t, \tag{49}$$

where $t > 0$ is a given constant. In particular, if $C = \{x \in \mathbb{R}^N : \|x\|_1 \leq t\}$ and $Q = \{y\}$, then the LASSO problem can be considered as the SFP. From this connection, we can apply the CQ algorithm to solve Equation (49).

In this example, the sparse vector $x \in \mathbb{R}^N$ is generated by the uniform distribution in $[-2, 2]$ with $m$ nonzero elements. The matrix $A$ is generated by the normal distribution with mean zero and invariance one. The observation $y$ is generated by the white Gaussian noise with SNR=40. The process is started with $t = m$ and initial point $x_1 = ones(N, 1)$.

The stopping criterion is defined by the mean square error (MSE):

$$E_n = \frac{1}{N} \|x_n - x^*\|^2 < \kappa, \tag{50}$$

where $x_n$ is an estimated signal of $x^*$ and $\kappa$ is a tolerance.

In what follows, let $\mu = 0.3$, $\sigma = 0.2$, $\rho = 0.4$, $\beta_n = 1.9$ and $\theta_n = \dfrac{1}{200n + 1}$. The numerical results are reported as follows.

In Table 1, we observe that the performance of Algorithm 4 is better than other algorithms in terms of CPU time and number of iterations as the spikes of sparse vector is varied from 10 to 30. In this example, it is shown that Algorithm 4 of Yang [6], for which the step size depends on the norm of $A$, converges more slowly than other algorithms in terms of CPU time.

Next, we provide Figures 1–3 to illustrate the convergence behavior, MSE, number of iterations and objective function values when $N = 1024$, $M = 512$, $m = 20$ and $\kappa = 10^{-5}$.

**Table 1.** Numerical results ($M = 512$ and $N = 1024$).

| **$m$-Sparse** | **Method** | **$\kappa = 10^{-4}$** | | **$\kappa = 10^{-5}$** | |
|---|---|---|---|---|---|
| | | **CPU** | **Iter** | **CPU** | **Iter** |
| $m = 10$ | Algorithm 1 | 0.7801 | 93 | 0.5931 | 83 |
| | Algorithm 2 | 0.0962 | 187 | 0.1000 | 158 |
| | Algorithm 3 | 0.1416 | 257 | 0.0605 | 74 |
| | Algorithm 4 | 0.0271 | 33 | 0.0592 | 39 |
| $m = 15$ | Algorithm 1 | 0.6345 | 93 | 0.6778 | 101 |
| | Algorithm 2 | 0.1020 | 196 | 0.1001 | 195 |
| | Algorithm 3 | 0.1087 | 170 | 0.0823 | 97 |
| | Algorithm 4 | 0.0251 | 35 | 0.0430 | 51 |
| $m = 20$ | Algorithm 1 | 1.1535 | 161 | 1.1177 | 156 |
| | Algorithm 2 | 0.1661 | 308 | 0.1573 | 296 |
| | Algorithm 3 | 0.3557 | 500 | 0.1139 | 134 |
| | Algorithm 4 | 0.0516 | 55 | 0.0695 | 78 |
| $m = 25$ | Algorithm 1 | 0.7380 | 103 | 2.9774 | 443 |
| | Algorithm 2 | 0.0990 | 196 | 0.4746 | 940 |
| | Algorithm 3 | 0.0623 | 115 | 0.7258 | 1308 |
| | Algorithm 4 | 0.0354 | 42 | 0.0922 | 165 |
| $m = 30$ | Algorithm 1 | 1.1423 | 168 | 3.7280 | 92 |
| | Algorithm 2 | 0.1568 | 321 | 1.7980 | 666 |
| | Algorithm 3 | 0.1219 | 164 | 0.4119 | 111 |
| | Algorithm 4 | 0.0704 | 70 | 0.1335 | 38 |

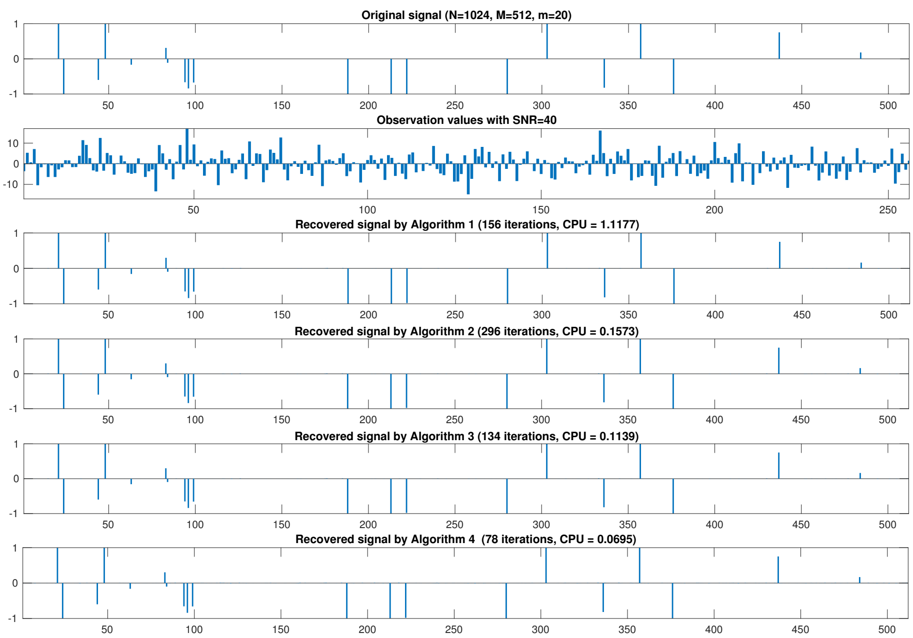


**Figure 1.** From top to bottom: original signal, observation data and recovered signal by Algorithms 1–4, respectively.

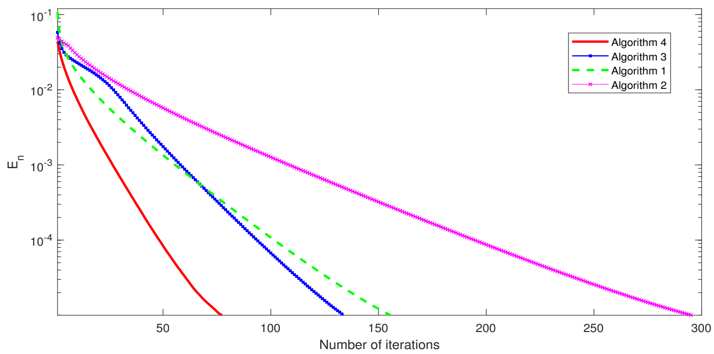


**Figure 2.** MSE versus number of iterations when $N = 1024$, $M = 512$ and $\kappa = 10^{-5}$.

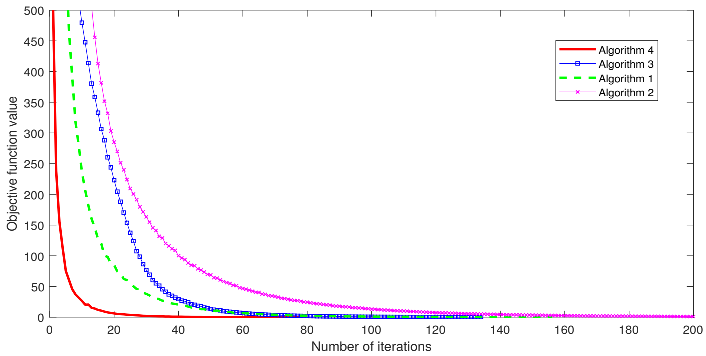


**Figure 3.** The objective function value versus number of iterations when $N = 1024$, $M = 512$ and $\kappa = 10^{-5}$.

In Figures 1–3, we can summarize that our proposed algorithm is really more efficient and faster than algorithms of Yang [6], Qu and Xiu [8] and Bnouhachem et al. [9].

In Table 2, we observe that Algorithm 4 is effective and also converges more quickly than Algorithm 1 of Yang [6], Algorithm 2 of Qu and Xiu [8] and Algorithm 3 of Bnouhuchem et al. [9]. Moreover, it is seen that Algorithm 1 of Yang [6] has the highest CPU time in computation. In this case, Algorithm 1 takes more CPU time than it does in the first case (see Table 1). Therefore, we can conclude that our proposed method has the advantage in comparison to other methods, especially Algorithm 1, which requires the spectral computation.

We next provide Figure 4–6 to illustrate the convergence behavior, MSE, number of iterations and objective function values when $N = 4096$, $M = 2048$, $m = 60$ and $\kappa = 10^{-5}$.

**Table 2.** Numerical results ($M = 2048$ and $N = 4096$).

| ***m*-sparse** | **Method** | $\kappa = 10^{-4}$ | | $\kappa = 10^{-5}$ | |
|---|---|---|---|---|---|
| | | **CPU** | **Iter** | **CPU** | **Iter** |
| $m = 20$ | Algorithm 1 | 53.4863 | 28 | 77.8192 | 40 |
| | Algorithm 2 | 3.1953 | 43 | 4.7627 | 62 |
| | Algorithm 3 | 1.5285 | 19 | 2.3102 | 28 |
| | Algorithm 4 | 1.0771 | 13 | 1.6199 | 20 |
| $m = 40$ | Algorithm 1 | 74.6456 | 38 | 106.3420 | 54 |
| | Algorithm 2 | 4.5607 | 60 | 6.1862 | 83 |
| | Algorithm 3 | 2.0701 | 26 | 2.9406 | 37 |
| | Algorithm 4 | 1.4418 | 18 | 2.1713 | 27 |
| $m = 60$ | Algorithm 1 | 86.1752 | 45 | 137.6885 | 70 |
| | Algorithm 2 | 5.2204 | 70 | 8.1821 | 110 |
| | Algorithm 3 | 2.3965 | 30 | 3.6434 | 46 |
| | Algorithm 4 | 1.7580 | 22 | 2.6908 | 34 |

**Table 2.** *Cont.*

| $m$-sparse | Method | $\kappa = 10^{-4}$ | | $\kappa = 10^{-5}$ | |
|---|---|---|---|---|---|
| | | CPU | Iter | CPU | Iter |
| $m = 80$ | Algorithm 1 | 133.5504 | 67 | 219.4587 | 112 |
| | Algorithm 2 | 7.8185 | 104 | 13.3599 | 178 |
| | Algorithm 3 | 3.4220 | 43 | 5.9392 | 75 |
| | Algorithm 4 | 2.4207 | 30 | 3.7902 | 47 |
| $m = 100$ | Algorithm 1 | 148.3098 | 75 | 327.4775 | 163 |
| | Algorithm 2 | 8.7840 | 118 | 19.7221 | 258 |
| | Algorithm 3 | 3.8024 | 48 | 16.0518 | 202 |
| | Algorithm 4 | 2.6962 | 34 | 5.3538 | 66 |

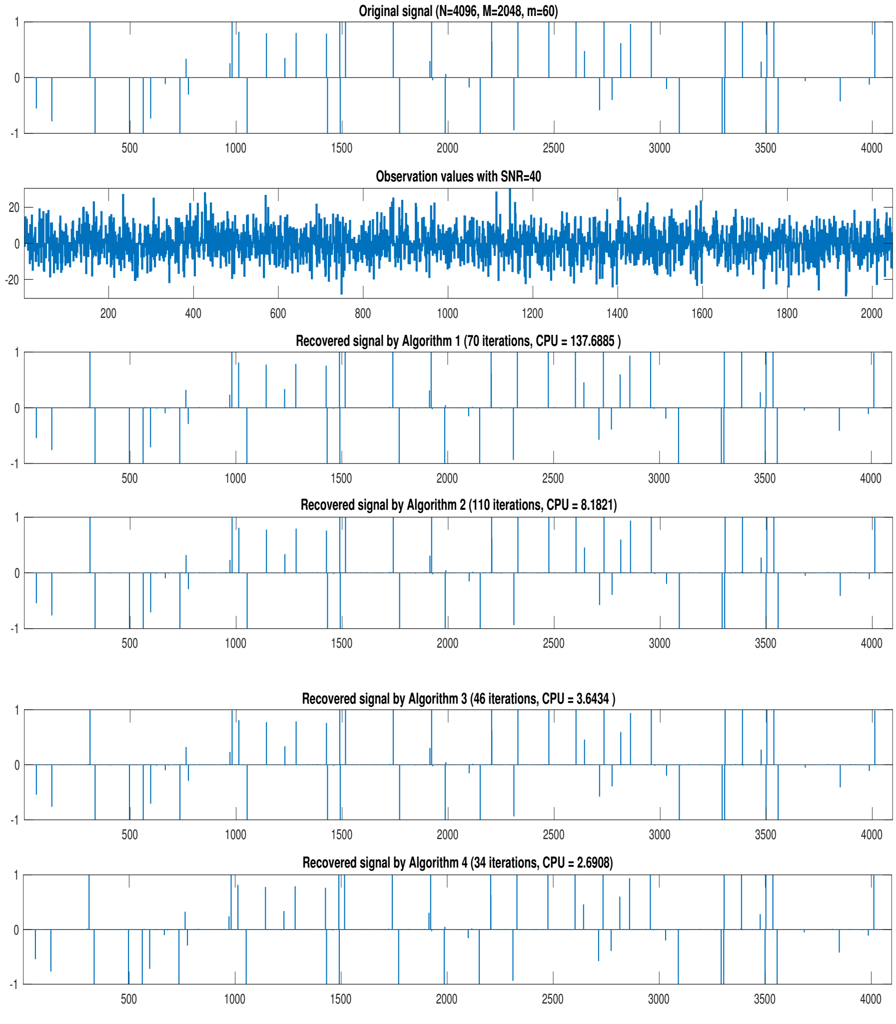


**Figure 4.** From top to bottom: original signal, observation data and recovered signal by Algorithms 1–4, respectively.

In Figures 4–6, we observe that MSE and objective function values of Algorithm 4 decreases faster than Algorithms 1–3 do in each cases.

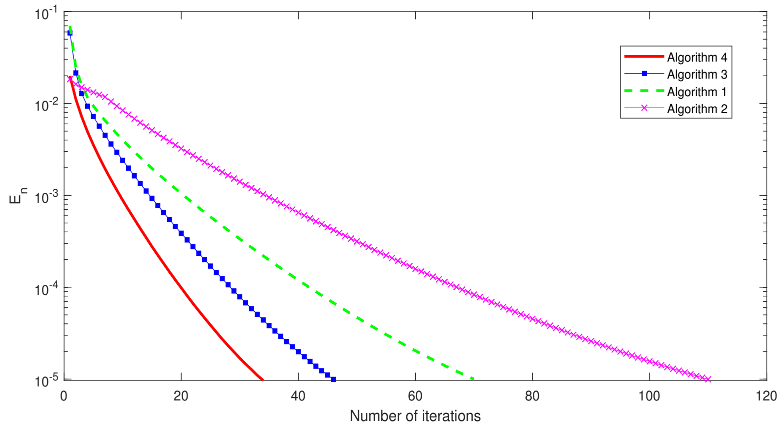


**Figure 5.** MSE versus number of iterations when $N = 4096$, $M = 2048$ and $\kappa = 10^{-5}$.

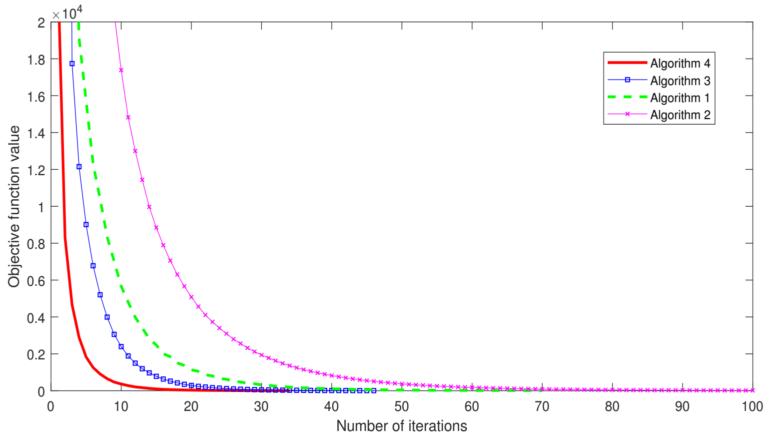


**Figure 6.** The objective function value versus number of iterations when $N = 4096$, $M = 2048$ and $\kappa = 10^{-5}$.

## 5. Comparative Analysis

In this section, we discuss the comparative analysis to show the effects of the step sizes $\alpha_n$ and $\beta_n$ in Algorithm 4.

We begin this section by studying the effect of the step size $\beta_n$ in Algorithm 4 in terms of the number of iterations and the CPU time with the varied cases.

Choose $\mu = 0.3$, $\sigma = 0.2$, $\rho = 0.4$ and $\theta_n = \frac{1}{200n+1}$. Let $x_1$ and $A$ be as in the previous example. The stopping criterion is defined by Equation (50) with $\kappa = 10^{-5}$.

**Table 3.** The convergence behavior of Algorithm 4 with different cases of $\beta_n$.

| | $\beta_n$ | CPU | Iter |
|---|---|---|---|
| $N = 1024$ | 0.1 | 0.4585 | 139 |
| $M = 512$ | 0.5 | 0.1976 | 73 |
| $m = 20$ | 1.0 | 0.1632 | 55 |
| | 1.5 | 0.1272 | 44 |
| | 2.0 | 0.1187 | 38 |
| | 2.5 | 0.1048 | 35 |
| | 3.0 | 0.1065 | 32 |
| | 3.5 | 0.1298 | 29 |
| | 3.9 | 0.0954 | 28 |

**Table 3.** *Cont.*

| | $\beta_n$ | CPU | Iter |
|---|---|---|---|
| $N$ = 4096 | 0.1 | 4.4547 | 58 |
| $M$ = 2048 | 0.5 | 3.6075 | 39 |
| $m$ = 20 | 1.0 | 2.2021 | 29 |
| | 1.5 | 1.8119 | 24 |
| | 2.0 | 1.6024 | 21 |
| | 2.5 | 1.5748 | 29 |
| | 3.0 | 1.4055 | 17 |
| | 3.5 | 1.3297 | 16 |
| | 3.9 | 1.3172 | 15 |

In Table 3, it is observed that the number of iterations and the CPU time have small reduction when the step size $\beta_n$ tends to 4. The numerical experiments for each cases of $\beta_n$ are shown in Figure 7 and Figure 8, respectively.

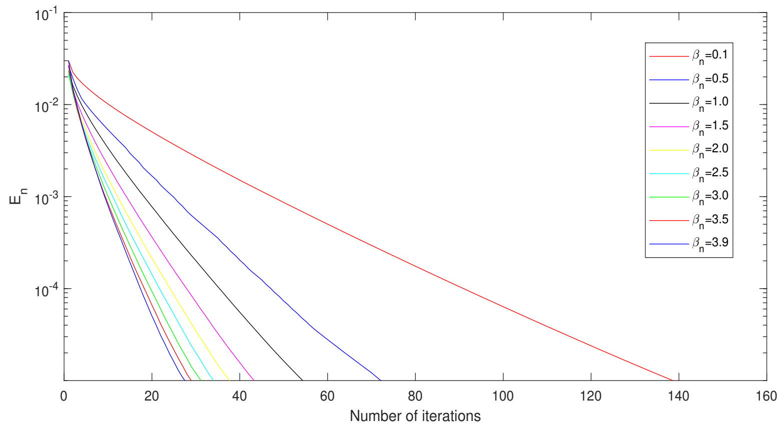


**Figure 7.** Graph of number of iterations versus $E_n$ in case $N = 1024$ and $M = 512$.

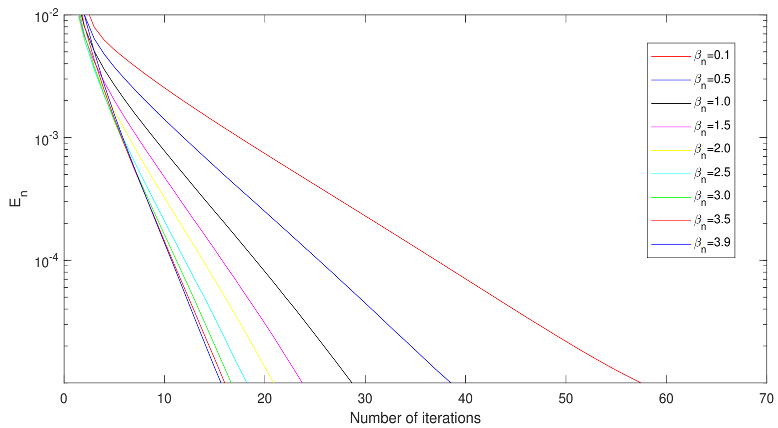


**Figure 8.** Graph of number of iterations versus $E_n$ in case $N = 4096$ and $M = 2048$.

Next, we discuss the effect of the step size $\alpha_n$ in Algorithm 4. We note that the step size $\alpha_n$ depends on the parameters $\rho$ and $\sigma$. Thus, we aim to vary these parameters and study its convergence behavior.

Choose $\mu = 0.3$, $\sigma = 0.2$, $\beta_n = 3.9$ and $\theta_n = \frac{1}{200n+1}$. Let $x_1$ and $A$ be as in the previous example. The stopping criterion is defined by Equation (50) with $\kappa = 10^{-5}$. The numerical results are reported in Table 4.

In Table 4, we see that the CPU time decreases significantly when the parameter $\rho$ is also decreased. However, the choice of $\rho$ has no effect in terms of number of iterations.

**Table 4.** The convergence behavior of Algorithm 4 with different cases of $\rho$.

| | $\rho$ | **CPU** | **Iter** |
|---|---|---|---|
| $N$ = 1024 | 0.1 | 0.0634 | 27 |
| $M$ = 512 | 0.3 | 0.0981 | 26 |
| $m$ = 20 | 0.5 | 0.1065 | 26 |
| | 0.7 | 0.1773 | 27 |
| | 0.9 | 0.5421 | 27 |
| $N$ = 4096 | 0.1 | 0.7554 | 17 |
| $M$ = 2048 | 0.3 | 1.2094 | 17 |
| $m$ = 20 | 0.5 | 1.7697 | 17 |
| | 0.7 | 3.1876 | 17 |
| | 0.9 | 10.1536 | 18 |

Next, we discuss the effect of $\sigma$ in Algorithm 4. In this experiment, choose $\mu = 0.3$, $\beta_n = 3.9$, $\rho = 0.5$ and $\theta_n = \frac{1}{200n+1}$. The error $E_n$ is defined by Equation (50) with $\kappa = 10^{-5}$. The numerical results are reported in Table 5.

**Table 5.** The convergence behavior Algorithm 4 with different cases of $\sigma$.

| | $\sigma$ | **CPU** | **Iter** |
|---|---|---|---|
| $N$ = 1024 | 1 | 0.2985 | 53 |
| $M$ = 512 | 2 | 0.2974 | 53 |
| $m$ = 20 | 3 | 0.2636 | 56 |
| | 4 | 0.2478 | 53 |
| | 5 | 0.2584 | 52 |
| | 6 | 0.2816 | 56 |
| $N$ = 4096 | 1 | 1.9105 | 16 |
| $M$ = 2048 | 2 | 1.9990 | 16 |
| $m$ = 20 | 3 | 2.0937 | 16 |
| | 4 | 2.1371 | 16 |
| | 5 | 2.2449 | 17 |
| | 6 | 2.3816 | 16 |

In Table 5, we observe that the choices of $\sigma$ have a small effect in both terms of the CPU time and the number of iterations.

Finally, we discuss the convergence of Algorithm 4 with different cases of $M$ and $N$. In this case, we set $\sigma = 1$, $\rho = 0.5$, $\mu = 0.3$, $\beta_n = 3.9$ and $\theta_n = \frac{1}{200n+1}$. The stopping criterion is defined by Equation (50).

In Table 6, it is shown that, if $M$ and $N$ have a high value, then the number of iteration decreases. However, in this case, the CPU time increases.

**Table 6.** The convergence behavior Algorithm 4 with different cases of $M$ and $N$.

| | $\kappa = 10^{-4}$ | | $\kappa = 10^{-5}$ | |
|---|---|---|---|---|
| | **CPU** | **Iter** | **CPU** | **Iter** |
| $M = 1024$ $N = 2048$ | 0.9967 | 13 | 1.4998 | 21 |
| $M = 2048$ $N = 4096$ | 3.8625 | 11 | 5.6119 | 16 |
| $M = 3072$ $N = 6144$ | 5.0449 | 6 | 6.5788 | 8 |
| $M = 4096$ $N = 8192$ | 7.3689 | 5 | 10.1838 | 7 |

## 6. Conclusions

In this work, we introduce a new hybrid CQ algorithm by using the self adaptive and the line-search techniques for the split feasibility problem in Hilbert spaces. This method can be viewed as a refinement and improvement of other CQ algorithms. Convergence analysis of the proposed method is proved under some suitable conditions. The numerical results show that our algorithm has a better convergence behavior than the algorithms of Yang [6], Qu and Xiu [8] and Bnouhachem et al. [9]. A comparative analysis was also performed to show the effects of the step sizes in our algorithm.

**Author Contributions:** S.S.; supervision and investigation, S.K.; writing original draft and P.C.; formal analysis and methodology.

**Funding:** This research was funded by Chiang Mai University.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Censor, Y.; Elfving, T. A multiprojection algorithms using Bregman projection in a product space. *Numer. Algor.* **1994**, *8*, 221–239. [CrossRef]
2. Bauschke, H.H.; Borwein, J.M. On projection algorithms for solving convex feasibility problems. *SIAM Rev.* **1996**, *38*, 367–426. [CrossRef]
3. Stark, H.; *Image Recovery: Theory and Application*; Elsevier: Amsterdam, The Netherlands, 2013.
4. Byrne, C. Iterative oblique projection onto convex sets and the split feasibility problem. *Inverse Probl.* **2002**, *18*, 441–453. [CrossRef]
5. Byrne, C. A unified treatment of some iterative algorithms in signal processing and image reconstruction. *Inverse Probl.* **2004**, *20*, 103–120. [CrossRef]
6. Yang, Q. The relaxed CQ algorithm for solving the split feasibility problem. *Inverse Prob.* **2004**, *20*, 1261–1266. [CrossRef]
7. López, G.; Martín-Márquez, V.; Wang, F.; Xu, H.K. Solving the split feasibility problem without prior knowledge of matrix norms. *Inverse Probl.* **2012**, *28*, 085004. [CrossRef]
8. Qu, B.; Xiu, N. A note on the CQ algorithm for the split feasibility problem. *Inverse Prob.* **2005**, *21*, 1655–1665. [CrossRef]
9. Bnouhachem, A.; Noor, M.A.; Khalfaoui, M.; Zhaohan, S. On descent-projection method for solving the split feasibility problems. *J. Glob. Optim.* **2012**, *54*, 627–639. [CrossRef]
10. Dong, Q.L.; Tang, Y.C.; Cho, Y.J.; Rassias, T.M. "Optimal" choice of the step length of the projection and contraction methods for solving the split feasibility problem. *J. Glob. Optim.* **2018**, *71*, 341–360. [CrossRef]

11. Gibali, A.; Liu, L.W.; Tang, Y.C. Note on the modified relaxation CQ algorithm for the split feasibility problem. *Optim. Lett.* **2017**, *12*, 1–14. [CrossRef]
12. Bauschke, H.H.; Combettes, P.L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*; Springer: London, UK, 2011.