Chapter 3

An Accelerated Forward-Backward Splitting Algorithm for Solving Inclusion Problems with Applications

The previous chapter is dedicated to develop accelerated fixed point technique, which is further used to solve the monotone inclusion problem. In this chapter, we propose a novel accelerated preconditioned forward-backward algorithm to obtain the vanishing point of the sum of two operators in which one is maximal monotone and the other is M-cocoercive, where M is a linear bounded operator on underlying spaces.

This chapter is based on our published research work "**Dixit, A.**, Sahu, D. R., Gautam, P., and Som, T. and Yao, J.C. (2021). An accelerated forward-backward splitting algorithm for solving inclusion problems with applications to regression and link prediction problems, J. Nonlinear Var. Anal. 5, 79-101."

Section 3.1 is introductory while section 3.2 explains the results useful to this chapter. In Section 3.3, we propose a preconditioned forward-backward algorithm and study its convergence behavior under mild restrictions on operators and parameters. We also discuss a numerical example in the support of our findings which shows that in the same environment the proposed algorithm has better convergence speed than the algorithm proposed by Lorenz and Pock [61]. In Section 3.4, we apply the proposed algorithm to solve the saddle point problem. In the last section, we perform numerical experiments to show the practicability of the proposed algorithm and compared its convergence speed with those of already known algorithms. We apply the proposed algorithm to solve regression problems and link prediction problems.

3.1 Introduction

In 2015, Lorenz and Pock [61] used a variable metric (or preconditioning) approach to solve the monotone inclusion problem (1.8). For a linear, bounded, self-adjoint and positive definite operator $M : \mathcal{H} \to \mathcal{H}$, the algorithm proposed by Lorenz and Pock [61] can be written as follows:

$$\begin{cases} y_n = x_n + \theta_n(x_n - x_{n-1}) \\ x_{n+1} = (Id + \lambda_n M^{-1}A)^{-1}(Id - \lambda_n M^{-1}B)(y_n), \;\; n \in \mathbb{N} \end{cases} \tag{3.1}$$

where $\theta_n \in [0, 1)$ is an acceleration parameter and λ_n is a step size parameter. They studied the convergence of the algorithm, which can be summarized in the following theorem.

Theorem 3.1.1. ([61]) *Let $\mathcal{H}$ be a real Hilbert space and $A, B : \mathcal{H} \to 2^{\mathcal{H}}$ be maximally monotone operators. Further, assume that $M, L : \mathcal{H} \to \mathcal{H}$ are linear bounded* *selfadjoint and positive definite operators and that B is single-valued and cocoercive with respect to L^{-1}. Moreover, let $\lambda_n > 0$, $\theta < 1, \theta_n \in [0, \theta]$, $x_0 = x_1 \in \mathcal{H}$. If*

(i) $S_n = M - \frac{\lambda_n}{2} L$ *is positive definite for all* n*;*

(ii) $\sum_{n=1}^{\infty} \theta_n \|x_n - x_{n-1}\|_M^2 < \infty$,

then the sequence $\{x_n\}$ *generated by Algorithm (3.1) converges weakly to a solution of the inclusion problem (1.8).*

The aim of this paper is in three folds. Our first aim is to propose a preconditioned forward-backward algorithm to solve the monotone inclusion problem (1.8). Since the second assumption in Theorem 3.1.1 is very strong, which is not easy to verify and reduces its practical applicability, so our second aim is to study the convergence behavior of the proposed algorithm with mild assumptions so that it can be useful in practical applicability. Lastly, we aim to show the application of the proposed algorithm to solve regression problems and link prediction problems.

3.2 Preliminary Results

This section is devoted to some important definitions and results from nonlinear analysis and operator theory. Let M be a linear bounded operator on $\mathcal{H}$. M is said to be self-adjoint if $M^* = M$, where M^* denotes the adjoint of operator M. A self-adjoint operator M on $\mathcal{H}$ is said to be positive definite if $\langle M(x), x\rangle > 0$ for every nonzero $x \in \mathcal{H}$ ([59]). Define the M-inner product $\langle \cdot, \cdot \rangle_M$ on $\mathcal{H}$ by $\langle x, y\rangle_M = \langle x, M(y)\rangle$ for all $x, y \in \mathcal{H}$. The corresponding M-norm is defined by $\|x\|_M^2 = \langle x, Mx\rangle$ for all $x \in \mathcal{H}$.

Definition 3.1. Let D be a nonempty subset of $\mathcal{H}$, $T : D \to \mathcal{H}$ be an operator and $M : \mathcal{H} \to \mathcal{H}$ be a positive definite operator. Then T is said to be

(i) nonexpansive with respect to M-norm if

$$\|Tx_1 - Tx_2\|_M \leq \|x_1 - x_2\|_M \ \forall x_1, x_2 \in \mathcal{H};$$

(ii) M-cocoercive if

$$\|Tx_1 - Tx_2\|_{M^{-1}}^2 \leq \langle x_1 - x_2, Bx_1 - Bx_2 \rangle, \text{ for all } x_1, x_2 \in \mathcal{H}.$$

Example 3.1. *Define* $B : \mathbb{R}^3 \to \mathbb{R}^3$ *by* $(x_1, x_2, x_3) \mapsto (5x_1, 4\sin x_2, tan^{-1}(5x_3))$ *and*

$$M = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

For $x = (x_1, x_2, x_3)$ *and* $y = (y_1, y_2, y_3)$, *we have*

$$\begin{aligned} \|Bx - By\|_{M^{-1}}^2 &= 5(x_1 - y_1)^2 + 4(\ sin\ x_2 - sin\ y_2)^2 + (tan^{-1}5x_3 - tan^{-1}5y_3)^2 \\ &= \langle x - y, Bx - By \rangle. \end{aligned}$$

Thus, B *is* M*-cocoerceive.*

Lemma 3.2.1. [9, Corollary 2.14] *Let* $z_1, z_2 \in \mathcal{H}$. *Then the following identities hold for arbitrary* $a \in \mathbb{R}$:

(i) $\|z_1 - z_2\|^2 = \|z_1\|^2 + \|z_2\|^2 - 2\langle z_1, z_2 \rangle$,

(ii) $\|az_1 + (1 - a)z_2\|^2 = a\|z_1\|^2 + (1 - a)\|z_2\|^2 - a(1 - a)\|z_1 - z_2\|^2$.

Lemma 3.2.2. *[37] Let ρ be positive and α be nonnegative real numbers. Then, for each $z_1,\ z_2 \in \mathcal{H}$,*

$$\|z_1 \pm \alpha z_2\|^2 \geq (1-\alpha\rho)\|z_1\|^2 + \alpha(\alpha - \frac{1}{\rho})\|z_2\|^2.$$

Lemma 3.2.3. [9, Corollary 4.18] *Let $\mathcal{C}$ be a nonempty closed convex subset of $\mathcal{H}$ and $T : \mathcal{C} \to \mathcal{H}$ be a nonexpansive mapping. Let $\{z_n\}$ be a sequence in $\mathcal{C}$ and $z \in \mathcal{H}$ be such that $z_n \rightharpoonup z$ and $z_n - Tz_n \to 0$ as $n \to \infty$. Then $z \in \operatorname{Fix}(T)$.*

Lemma 3.2.4. [5, Lemma 3] *Consider sequences $\{y_n\}, \{z_n\}$ and $\{\theta_n\}$ in $[0, \infty)$ such that*

$$y_{n+1} \leq y_n + \theta_n(y_n - y_{n-1}) + z_n \ \ \textit{for all} \ \ n \in \mathbb{N}, \quad \sum_{n=1}^{\infty} z_n < \infty$$

and there exists a real number θ with $0 \leq \theta_n \leq \theta < 1$ for all $n \in \mathbb{N}$. Then the following hold:

(i) $\sum_{n=1}^{\infty}[y_n - y_{n-1}]_+ < \infty$, *where* $[t]_+ = \max\{t, 0\}$,

(ii) *there exists* $y^* \in [0, \infty)$ *such that* $y_n \to y^*$.

Lemma 3.2.5. *[74] Consider a nonempty subset $\mathcal{C}$ of $\mathcal{H}$. Let $\{\phi_n\}$ be a sequence in $\mathcal{H}$ such that the following two conditions hold:*

(i) *for all* $\phi \in \mathcal{C}$, $\lim_{n\to\infty} \|\phi_n - \phi\|$ *exists,*

(ii) *every sequential weak cluster point of* $\{\phi_n\}$ *is in* $\mathcal{C}$.

Then the sequence $\{\phi_n\}$ converges weakly to a point in $\mathcal{C}$.

3.3 Main Results

Throughout this section, we study the monotone inclusion problem (1.8) where $A : \mathcal{H} \to 2^{\mathcal{H}}$ is a maximal monotone operator and $B : \mathcal{H} \to \mathcal{H}$ is M-cocoerecive. Let $M : \mathcal{H} \to \mathcal{H}$ be a linear selfadjoint and positive definite operator. For $\lambda \in (0, \infty)$, define an operator $J^{A,B}_{\lambda,M}$ by

$$J^{A,B}_{\lambda,M} = (Id + \lambda M^{-1}A)^{-1}(Id - \lambda M^{-1}B). \tag{3.2}$$

We now give some properties of the operator $J^{A,B}_{\lambda,M}$.

Proposition 3.2. *Let $A : \mathcal{H} \to 2^{\mathcal{H}}$ be a maximal monotone operator and $M : \mathcal{H} \to \mathcal{H}$ be a linear bounded selfadjoint and positive definite operator. Let $B : \mathcal{H} \to \mathcal{H}$ be M-cocoerecive operator and $\lambda \in (0, 1]$. Then we have the following:*

(a) $Id - \lambda M^{-1}B$ is nonexpansive with respect to M-norm.

(b) $(Id + \lambda M^{-1}A)^{-1}$ is nonexpansive with respect to M-norm.

(c) The operator $J^{A,B}_{\lambda,M}$ defined by (3.2) is nonexpansive with respect to M-norm.

Proof. (a) Define $S = Id - M^{-1}B$. We show that S is nonexpansive with respect to M-norm. Let $x, y \in \mathcal{H}$. Since B is M-cocoerecive, we have

$$\|Bx - By\|^2_{M^{-1}} \leq \langle x - y, Bx - By\rangle \leq 2\langle x - y, Bx - By\rangle.$$

Thus,

$$\|M^{-1}(Bx - By)\|^2_M = \|Bx - By\|^2_{M^{-1}} \leq 2\langle x - y, Bx - By\rangle,$$

which implies that

$$\lambda^2\|M^{-1}(Bx-By)\|_M^2 \leq 2\langle x-y, \lambda M^{-1}(Bx-By)\rangle_M.$$

Hence

$$\|x-y\|_M^2 + \|\lambda M^{-1}(Bx-By)\|_M^2 - 2\langle x-y, \lambda M^{-1}(Bx-By)\rangle_M \leq \|x-y\|_M^2,$$

i.e,

$$\|(Id-\lambda M^{-1}B)x - (Id-\lambda M^{-1}B)y\|_M^2 \leq \|x-y\|_M^2.$$

Therefore, S is nonexpansive with respect to M-norm.

(b) Define $T = (Id + \lambda M^{-1}A)^{-1}$. Note that

$$T = (Id + \lambda M^{-1}A)^{-1} \Leftrightarrow T^{-1} - Id = \lambda M^{-1}A \Leftrightarrow \lambda A = M(T^{-1} - Id).$$

Let $x, y \in \mathcal{H}$. Then $M(x - T(x)) \in \lambda A(T(x))$ and $M(y - T(y)) \in \lambda A(T(y))$. Since A is monotone, we have

$$\langle Tx - Ty, M(x - Tx) - M(y - Ty)\rangle \geq 0,$$

i.e.,

$$\langle Tx - Ty, M(x-y)\rangle \geq \langle Tx - Ty, M(Tx - Ty)\rangle.$$

Then

$$\langle Tx - Ty, (x-y)\rangle_M \geq \|Tx - Ty\|_M^2.$$

Thus, T is nonexpansive with respect to M-norm.

(c) From (a) and (b), we see that the operator $T \circ S = J_{\lambda,M}^{A,B}$ is nonexpansive with respect to M-norm. □

Proposition 3.3. *Let $A : \mathcal{H} \to 2^{\mathcal{H}}$ be a maximal monotone operator, $M : \mathcal{H} \to \mathcal{H}$ be a linear bounded selfadjoint and positive definite operator, and $B : \mathcal{H} \to \mathcal{H}$ be an M-cocoerecive operator. Let $\lambda \in (0, \infty)$. Then $x^* \in \mathcal{H}$ is a solution of inclusion problem (1.8) if and only if x^* is the fixed point of the operator $J_{\lambda,M}^{A,B}$.*

Proof. Suppose that $0 \in (A + B)(x^*)$. Then

$$\begin{aligned} 0 \in \lambda A(x^*) + \lambda B(x^*) \;&\Leftrightarrow\; 0 \in \lambda M^{-1}A(x^*) + \lambda M^{-1}B(x^*) \\ &\Leftrightarrow\; -\lambda M^{-1}B(x^*) \in \lambda M^{-1}A(x^*) \\ &\Leftrightarrow\; x^* - \lambda M^{-1}B(x^*) \in x^* + \lambda M^{-1}A(x^*) \\ &\Leftrightarrow\; x^* = (Id + \lambda M^{-1}A)^{-1}(Id - \lambda M^{-1}B)(x^*). \end{aligned}$$

□

From Proposition 3.3, we conclude that inclusion problem (1.8) can be solved by finding fixed points of the operator $J_{\lambda,M}^{A,B}$. In the light of this fact and motivated by [37], we propose the following algorithm.

Algorithm 3.3.1. *Let $A : \mathcal{H} \to 2^{\mathcal{H}}$ be a maximal monotone operator, $M : \mathcal{H} \to \mathcal{H}$ a linear selfadjoint and positive definite operator and $B : \mathcal{H} \to \mathcal{H}$ an M-cocoerecive operator. Let $x_0, x_1 \in \mathcal{H}$. The accelerated preconditioning forward-backward normal*

S-iteration method (APFBNSM) is defined as follows:

$$\begin{cases} y_n = x_n + \alpha_n(x_n - x_{n-1}) \\ x_{n+1} = J^{A,B}_{\lambda,M}[(1-\beta_n)y_n + \beta_n J^{A,B}_{\lambda,M}(y_n)], \quad \textit{for all } n \in \mathbb{N}, \end{cases} \tag{3.3}$$

where $\beta_n \in (0,1), \lambda \in (0,1], \alpha_n \in [0,1)$.

For $\alpha_n = 0$, $\forall n \in \mathbb{N}$ and $M = Id$, the accelerated preconditioning forward-backward normal S-iteration method (3.3) is reduced to the normal S-iteration based forward-backward splitting algorithm (nS-FBSA) ([89]):

$$\begin{cases} y_n = x_n + \theta_n(x_n - x_{n-1}) \\ x_{n+1} = J^{A,B}_{\lambda}[(1-\beta_n)y_n + \beta_n J^{A,B}_{\lambda}(y_n)], \quad \text{for all } n \in \mathbb{N}. \end{cases}$$

Assumption 3.3.1.

Consider the parameters $\alpha_n, \beta_n, \lambda$ *satisfying the following conditions:*

(B1) $\{\alpha_n\} \subset [0,\alpha]$ *is a non-decreasing sequence with* $\alpha \in [0,1)$*;*

(B2) $\{\beta_n\} \subset (0,1)$ *and* $\lambda \in (0,1]$*;*

(B3) constants $\beta,\ \tau,\ \delta > 0$ *satisfying*

$$\delta > \frac{2\gamma\alpha(\alpha(1+\alpha)+\tau)}{1-\alpha^2(1-\beta)} \textit{ and } 0 < \beta \leq \beta_n \leq \frac{\delta - \alpha(2\gamma\alpha(1+\alpha)+\alpha\delta(1-\beta)+2\gamma\tau)}{\delta[1+2\gamma\alpha(1+\alpha)+\alpha\delta(1-\beta)+2\gamma\tau]},$$

where $\gamma = 1 + \frac{1}{\beta^2}$.

3.3.1 Convergence analysis of the APFBNSM

Proposition 3.3.1. *Let $M : \mathcal{H} \to \mathcal{H}$ be a linear bounded selfadjoint and positive definite operator. Let $A : \mathcal{H} \to 2^{\mathcal{H}}$ be a maximally monotone and $B : H \to H$ be M-cocoercive operator such that $(A+B)^{-1}(0)$ is nonempty. Assume that λ and sequences $\{\alpha_n\}$ and $\{\beta_n\}$ satisfy Assumption 3.3.1. Let $x^* \in (A+B)^{-1}(0)$ and $\{x_n\}$ be a sequence in H generated by Algorithm 3.3.1. Then*

$$\begin{aligned}\|x_{n+1} - x^*\|_M^2 \quad &\leq \quad (1+\alpha_n)\|x_n - x^*\|_M^2 - \alpha_n\|x_{n-1} - x^*\|_M^2 + \alpha_n(1+\alpha_n)\|x_n - x_{n-1}\|_M^2 \\ &\quad -\beta_n(1-\beta_n)\|y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2 \quad \textit{for all} \quad n \in \mathbb{N}.\end{aligned}$$

Proof. From Algorithm 3.3.1 and Lemma 3.2.1, we have

$$\begin{aligned}\|x_{n+1} - x^*\|_M^2 &= \|J_{\lambda,M}^{A,B}[(1-\beta_n)y_n + \beta_n J_{\lambda,M}^{A,B}(y_n)] - x^*\|_M^2 \\ &\leq \|(1-\beta_n)y_n + \beta_n J_{\lambda,M}^{A,B}(y_n) - x^*\|_M^2 \\ &= (1-\beta_n)\|y_n - x^*\|_M^2 + \beta_n\|J_{\lambda,M}^{A,B}(y_n) - x^*\|_M^2 - \beta_n(1-\beta_n)\|y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2 \\ &\leq (1-\beta_n)\|y_n - x^*\|_M^2 + \beta_n\|y_n - x^*\|_M^2 - \beta_n(1-\beta_n)\|y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2 \\ &= \|y_n - x^*\|_M^2 - \beta_n(1-\beta_n)\|y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2. \qquad (3.4)\end{aligned}$$

Again, from (3.3.1) and Lemma 3.2.1, we have

$$\begin{aligned}\|y_n - x^*\|_M^2 &= \|x_n + \alpha_n(x_n - x_{n-1}) - x^*\|_M^2 \\ &= \|(1+\alpha_n)(x_n - x^*) - \alpha_n(x_{n-1} - x^*)\|_M^2 \\ &= (1+\alpha_n)\|x_n - x^*\|_M^2 - \alpha_n\|x_{n-1} - x^*\|_M^2 + \alpha_n(1+\alpha_n)\|x_n - x_{n-1}\|_M^2.\end{aligned}$$

(3.5)

Combining (3.4) and (3.5), we have

$$\begin{aligned} \|x_{n+1}-x^*\|_M^2 &\leq (1+\alpha_n)\|x_n-x^*\|_M^2-\alpha_n\|x_{n-1}-x^*\|_M^2+\alpha_n(1+\alpha_n)\|x_n-x_{n-1}\|_M^2 \\ &\quad -\beta_n(1-\beta_n)\|y_n-J_{\lambda,M}^{A,B}(y_n)\|_M^2. \end{aligned}$$

This completes the proof. □

Define sequences $\{\mu_n\}$ and $\{\xi_n\}$ by

$$\mu_n=\alpha_n(1+\alpha_n)+\frac{\alpha_n(1-\beta_n)(1-\alpha_n\rho_n)}{2\gamma\beta_n\rho_n} \text{ and } \xi_n=\frac{(1-\beta_n)(\alpha_n\rho_n-1)}{2\gamma\beta_n}, \tag{3.6}$$

where $\rho_n=\frac{1}{\alpha_n+\delta\beta_n}$.

Proposition 3.3.2. *Let $M:\mathcal{H}\to\mathcal{H}$ be a linear bounded selfadjoint and positive definite operator. Let $A:\mathcal{H}\to 2^{\mathcal{H}}$ be a maximally monotone and let $B:H\to H$ be M-cocoercive operator such that $(A+B)^{-1}(0)$ is nonempty. Assume that λ and sequences $\{\alpha_n\}$ and $\{\beta_n\}$ satisfy Assumption 3.3.1. Let $x^*\in(A+B)^{-1}(0)$ and $\{x_n\}$ be a sequence in H generated by Algorithm 3.3.1. Then*

$$\phi_{n+1}-(1+\alpha_n)\phi_n+\alpha_n\phi_{n-1}\leq\xi_n\|x_{n+1}-x_n\|_M^2+\mu_n\|x_n-x_{n-1}\|_M^2, n\in\mathbb{N}, \tag{3.7}$$

where $\phi_n=\|x_n-x^*\|_M^2$.

Proof. Set $z_n=(1-\beta_n)y_n+\beta_nJ_{\lambda,M}^{A,B}(y_n)$. Then Algorithm 3.3.1 can be written as:

$$\begin{cases} y_n=x_n+\alpha_n(x_n-x_{n-1}) \\ z_n=(1-\beta_n)y_n+\beta_nJ_{\lambda,M}^{A,B}(y_n) \\ x_{n+1}=J_{\lambda,M}^{A,B}(z_n). \end{cases} \tag{3.8}$$

From (3.8), we have

$$\begin{aligned} \|y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2 &= \frac{1}{\beta_n^2}\|z_n - y_n\|_M^2 \\ &\geq \frac{1}{\beta_n^2}\|J_{\lambda,M}^{A,B}(z_n) - J_{\lambda,M}^{A,B}(y_n)\|_M^2 \\ &= \frac{1}{\beta_n^2}\|x_{n+1} - J_{\lambda,M}^{A,B}(y_n)\|_M^2 \\ &= \frac{1}{\beta_n^2}\|x_{n+1} - y_n + y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2. \end{aligned}$$

Taking $\rho = \frac{1}{2}$ and using Lemma 3.2.2, we obtain

$$\|y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2 \geq \frac{1}{\beta_n^2}\left\{\frac{1}{2}\|x_{n+1} - y_n\|_M^2 - \|y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2\right\},$$

which implies that

$$(1 + \frac{1}{\beta_n^2})\|y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2 \geq \frac{1}{2\beta_n^2}\|x_{n+1} - y_n\|_M^2.$$

Since β_n is bounded below by β, we have

$$\begin{aligned} (1 + \frac{1}{\beta^2})\|y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2 &\geq \frac{1}{2\beta_n^2}\|x_{n+1} - y_n\|_M^2 \\ &= \frac{1}{2\beta_n^2}\|x_{n+1} - x_n - \alpha_n(x_n - x_{n-1})\|_M^2. \end{aligned}$$

Again, using Lemma 3.2.2, we obtain

$$\begin{aligned} (1 + \frac{1}{\beta^2})\|y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2 &\geq \frac{(1 - \alpha_n\rho_n)}{2{\beta_n}^2}\|x_{n+1} - x_n\|_M^2 + \frac{\alpha_n}{2\beta_n^2}\left(\alpha_n - \frac{1}{\rho_n}\right)\|x_n - x_{n-1}\|_M^2 \\ &= \frac{(1 - \alpha_n\rho_n)}{2\beta_n^2}\|x_{n+1} - x_n\|_M^2 - \frac{\alpha_n(1 - \alpha_n\rho_n)}{2\beta_n^2\rho_n}\|x_n - x_{n-1}\|_M^2. \end{aligned} \tag{3.9}$$

Multiplying (3.9) by $-\beta_n(1-\beta_n)$, we obtain

$$
\begin{aligned}
-\gamma\beta_n(1-\beta_n)\|y_n - J_{\lambda,M}^{A,B}(y_n)\|_M^2 \quad &\le \quad -\frac{(1-\beta_n)(1-\alpha_n\rho_n)}{2\beta_n}\|x_{n+1}-x_n\|_M^2 \\
&\quad +\frac{\alpha_n(1-\beta_n)(1-\alpha_n\rho_n)}{2\beta_n\rho_n}\|x_n - x_{n-1}\|_M^2.
\end{aligned} \tag{3.10}
$$

From Proposition 3.3.1 and (3.10), we get

$$
\begin{aligned}
\|x_{n+1}-x^*\|_M^2 \quad &\le \quad (1+\alpha_n)\|x_n - x^*\|_M^2 - \alpha_n\|x_{n-1}-x^*\|_M^2 + \alpha_n(1+\alpha_n)\|x_n - x_{n-1}\|_M^2 \\
&\quad -\frac{(1-\beta_n)(1-\alpha_n\rho_n)}{2\gamma\beta_n}\|x_{n+1}-x_n\|_M^2 + \frac{\alpha_n(1-\beta_n)(1-\alpha_n\rho_n)}{2\gamma\beta_n\rho_n}\|x_n - x_{n-1}\|_M^2.
\end{aligned}
$$

Hence

$$
\begin{aligned}
\phi_{n+1} \quad &\le \quad (1+\alpha_n)\phi_n - \alpha_n\phi_{n-1} + \alpha_n(1+\alpha_n)\|x_n - x_{n-1}\|_M^2 \\
&\quad -\frac{(1-\beta_n)(1-\alpha_n\rho_n)}{2\gamma\beta_n}\|x_{n+1}-x_n\|_M^2 + \frac{\alpha_n(1-\beta_n)(1-\alpha_n\rho_n)}{2\gamma\beta_n\rho_n}\|x_n - x_{n-1}\|_M^2,
\end{aligned}
$$

which can be written as

$$
\phi_{n+1} - (1+\alpha_n)\phi_n + \alpha_n\phi_{n-1} \le \xi_n\|x_{n+1}-x_n\|_M^2 + \mu_n\|x_n - x_{n-1}\|_M^2.
$$

□

Proposition 3.3.3. *Suppose that $\{\alpha_n\}$ and $\{\beta_n\}$ are sequences in $[0,1)$ satisfying Assumption 3.3.1. Let $\{\xi_n\}$ and $\{\mu_n\}$ be sequences defined by (3.6). Then $\xi_n + \mu_{n+1} \le -\tau$ for all $n \in \mathbb{N}$.*

Proof. Observe that

$$\mu_n = \alpha_n(1+\alpha_n) + \frac{\alpha_n(1-\beta_n)(1-\alpha_n\rho_n)}{2\gamma\beta_n\rho_n} > 0,$$

since $\alpha_n\rho_n < 1$ and $\beta_n \in (0,1)$. Again, taking into account of choice of ρ_n, we have

$$\delta = \frac{1-\alpha_n\rho_n}{\rho_n\beta_n}.$$

Note

$$\mu_n = \alpha_n(1+\alpha_n) + \frac{\alpha_n(1-\beta_n)\delta}{2\gamma} \leq \alpha(1+\alpha) + \frac{\alpha\delta(1-\beta)}{2\gamma} \quad \text{for all } n \in \mathbb{N}. \tag{3.11}$$

For all $n \in \mathbb{N}$, we have

$$\begin{aligned}
\xi_n + \mu_{n+1} \leq -\tau \quad &\Leftrightarrow \quad \frac{(1-\beta_n)(\alpha_n\rho_n - 1)}{2\gamma\beta_n} + (\mu_{n+1}+\tau) \leq 0 \\
&\Leftrightarrow \quad (1-\beta_n)(\alpha_n\rho_n - 1) + 2\gamma\beta_n(\mu_{n+1}+\tau) \leq 0 \\
&\Leftrightarrow \quad -(1-\beta_n)\delta\rho_n\beta_n + 2\gamma\beta_n(\mu_{n+1}+\tau) \leq 0 \\
&\Leftrightarrow \quad -\frac{(1-\beta_n)\delta}{\alpha_n + \delta\beta_n} + 2\gamma(\mu_{n+1}+\tau) \leq 0 \\
&\Leftrightarrow \quad -(1-\beta_n)\delta + 2\gamma(\mu_{n+1}+\tau)(\alpha_n + \delta\beta_n) \leq 0 \\
&\Leftrightarrow \quad 2\gamma(\mu_{n+1}+\tau)(\alpha_n + \delta\beta_n) + \beta_n\delta \leq \delta.
\end{aligned}$$

By using (3.11), we have

$$\begin{aligned}
2\gamma(\mu_{n+1}+\tau)(\alpha_n+\delta\beta_n) + \beta_n\delta &\leq 2\gamma(\alpha(1+\alpha) + \frac{\alpha\delta(1-\beta)}{2\gamma} + \tau)(\alpha + \delta\beta_n) + \beta_n\delta \\
&\leq \delta,
\end{aligned}$$

where the last inequality follows by using the upper bound of $\{\beta_n\}$ in Assumption 3.3.1. Hence

$$\xi_n + \mu_{n+1} \leq -\tau \text{ for all } n \in \mathbb{N}.$$

□

Now, we establish the weak convergence of the accelerated preconditioning forward-backward normal S-iteration method (APFBNSM) defined by Algorithm 3.3.1 for the computation of solutions of inclusion problem (1.8).

Theorem 3.3.1. *Let $M : \mathcal{H} \to \mathcal{H}$ be a linear bounded selfadjoint and positive definite operator. Let $A : \mathcal{H} \to 2^{\mathcal{H}}$ be a maximally monotone and $B : \mathcal{H} \to \mathcal{H}$ be M-cocoercive operator such that $(A+B)^{-1}(0)$ is nonempty. Let $\lambda \in (0,1]$ and let $\{\alpha_n\}$ and $\{\beta_n\}$ satisfy the Assumption 3.3.1 with $\alpha_1 = 0$. Then the sequence $\{x_n\}$ generated by Algorithm 3.3.1 converges weakly to a point of $(A+B)^{-1}(0)$.*

Proof. Let $x^* \in (A+B)^{-1}(0)$. Set $\psi_n = \phi_n - \alpha_n \phi_{n-1} + \mu_n \|x_n - x_{n-1}\|_M^2$. We proceed with the following steps.

Step 1. $\sum_{n=1}^{\infty} \|x_{n+1} - x_n\|_M^2 < \infty$.

Consider

$$\begin{aligned}
\psi_{n+1} - \psi_n &= \phi_{n+1} - \alpha_{n+1}\phi_n + \mu_{n+1}\|x_{n+1} - x_n\|_M^2 - \phi_n + \alpha_n \phi_{n-1} - \mu_n \|x_n - x_{n-1}\|_M^2 \\
&= \phi_{n+1} - (1+\alpha_{n+1})\phi_n + \alpha_n \phi_{n-1} + \mu_{n+1}\|x_{n+1} - x_n\|_M^2 - \mu_n \|x_n - x_{n-1}\|_M^2.
\end{aligned}$$

Using Propositions 3.3.2 and 3.3.3, we have

$$\begin{aligned}\psi_{n+1}-\psi_n &\leq \xi_n\|x_{n+1}-x_n\|_M^2+\mu_{n+1}\|x_{n+1}-x_n\|_M^2\\ &= (\xi_n+\mu_{n+1})\|x_{n+1}-x_n\|_M^2.\\ &\leq -\tau\|x_{n+1}-x_n\|_M^2 \text{ for all } n\in\mathbb{N}, \end{aligned} \tag{3.12}$$

which implies that $\{\psi_n\}$ is nonincreasing sequence. Since $\{\alpha_n\}$ is bounded above by α, we obtain

$$-\alpha\phi_{n-1}\leq\phi_n-\alpha\phi_{n-1}\leq\psi_n\leq\psi_1.$$

Thus,

$$\begin{aligned}\phi_n &\leq \alpha\phi_{n-1}+\psi_1\\ &\leq \alpha(\alpha\phi_{n-2}+\psi_1)+\psi_1\\ &\vdots\\ &\leq \alpha^n\phi_0+\psi_1\sum_{k=0}^{n-1}\alpha^k\leq\alpha^n\phi_0+\frac{\psi_1}{1-\alpha}.\end{aligned}$$

From (3.12), we conclude that

$$\begin{aligned}\tau\sum_{k=1}^{n}\|x_{k+1}-x_k\|_M^2 &\leq \psi_1-\psi_{n+1}\\ &\leq \psi_1+\alpha\phi_n\\ &\leq \psi_1+\alpha(\alpha^n\phi_0+\frac{\psi_1}{1-\alpha})\\ &= \alpha^{n+1}\phi_0+\frac{\psi_1}{1-\alpha}.\end{aligned}$$

Since $\alpha^{n+1} \to 0$ as $n \to \infty$, we obtain that

$$\sum_{n=0}^{\infty} \|x_{n+1} - x_n\|_M^2 < \infty. \tag{3.13}$$

Step 2. $\lim\limits_{n\to\infty} ||x_n - x^*||_M$ exists.

From (3.7), (3.11), (3.13) and Lemma 3.2.4, we obtain that $\lim_{n\to\infty} \|x_n - x^*\|_M$ exists.

Step 3. Every sequential weak cluster point of sequence $\{x_n\}$ is in $\text{Fix}(J_{\lambda,M}^{A,B})$.

From (3.13), we have

$$\lim_{n\to\infty} \|x_n - x_{n-1}\|_M = 0. \tag{3.14}$$

From Algorithm 3.3.1, we have

$$\|y_n - x_{n+1}\|_M \leq \|x_n - x_{n+1}\|_M + \alpha\|x_n - x_{n-1}\|_M.$$

Using (3.14), we get that $\lim\limits_{n\to\infty} \|y_n - x_{n+1}\|_M = 0$. From (3.3), we have

$$\begin{aligned}
\|J_{\lambda,M}^{A,B} y_n - y_n\|_M &= \|J_{\lambda,M}^{A,B} y_n - x_{n+1} + x_{n+1} - y_n\|_M \\
&\leq \|J_{\lambda,M}^{A,B} y_n - x_{n+1}\|_M + \|x_{n+1} - y_n\|_M \\
&= \|J_{\lambda,M}^{A,B} y_n - J_{\lambda,M}^{A,B} z_n\|_M + \|x_{n+1} - y_n\|_M \\
&\leq \|y_n - z_n\|_M + \|x_{n+1} - y_n\|_M \\
&= \|y_n - (1-\beta_n)y_n - \beta_n J_{\lambda,M}^{A,B} y_n\|_M + \|x_{n+1} - y_n\|_M \\
&= \beta_n \|y_n - J_{\lambda,M}^{A,B} y_n\|_M + \|x_{n+1} - y_n\|_M,
\end{aligned}$$

which implies that

$$(1-\beta_n)\|J_{\lambda,M}^{A,B}y_n - y_n\|_M \leq \|x_{n+1}-y_n\|_M. \tag{3.15}$$

From (3.15), we obtain

$$\|J_{\lambda,M}^{A,B}y_n - y_n\|_M \to 0 \text{ as } n \to \infty. \tag{3.16}$$

Suppose that $\{x_n\}$ has a weak cluster point $x \in \mathcal{H}$. From step 2, $\{x_n\}$ has a subsequence $\{x_{n_k}\}$, which converges weakly to x. Using (3.16) and Lemma 3.2.3 for $\{x_{n_k}\}$, we have $x \in (A+B)^{-1}(0)$. It follows from Lemma 3.2.5 that $\{x_n\}$ converges weakly to a point in $(A+B)^{-1}(0)$. □

Remark 3.3.1. *In order to deal with the convergence of Algorithm 3.3.1, we assume that $\alpha_1 = 0$ in Theorem 3.3.1. We can obtain the same conclusion of Theorem 3.3.1 if we assume $x_1 = x_0$.*

3.3.2 Numerical comparison of Algorithms (3.1) and 3.3.1

The aim of numerical example is to study the convergence behavior of Algorithm 3.3.1 to solve the inclusion problem and compare its performance with Algorithm (3.1).

Let $\mathcal{H} = \mathbb{R}^3$ with Euclidean norm and $A : \mathbb{R}^3 \to \mathbb{R}^3$ be an operator defined by

$$A(x_1, x_2, x_3) = (x_2 - x_3, x_3 - x_1, x_1 - x_2) \text{ for all } (x_1, x_2, x_3) \in \mathbb{R}^3.$$

Then operator A is maximally monotone. Consider the operators B and M as in Example 3.1. The operator B is M-cocoerceive. Thus, we can apply Algorithms (3.1) and 3.3.1 to find the zeros of $A+B$. We choose initial points $x_1 = x_0 = (15, 15, 14)$, $\alpha_n = \frac{1}{20}$ and $\beta_n = 0.5$. We perform the experiment for 70 iterations with difference of norms of two consecutive values is taken to be less than 0.001 as the stopping criterion. The graph is plotted between the Euclidean norm of x_n and the number of iterations.

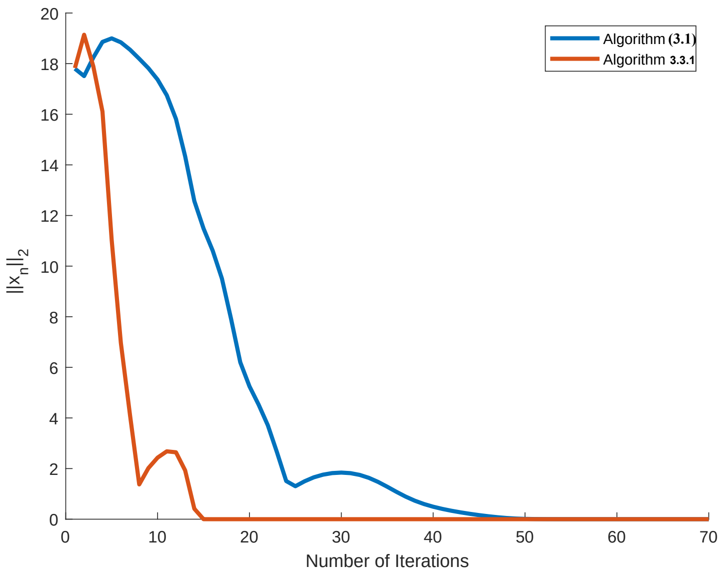

FIGURE 3.1: Behaviour of $\|x_n\|_2$ with respect to number of iterations .

In Figure 3.1, we can observe that $\|x_n\|_2$ corresponding to Algorithm 3.3.1 approaches towards 0 as the number of iterations increases, which supports the result proved in Theorem 3.3.1. It can also observe that graph of Algorithm (3.1) also converges to 0. From Figure 3.1, we can say that the convergence speed of Algorithm 3.3.1 is faster than Algorithm (3.1). Table 3.1 shows that in order to get value of $\|x_n\|_2$ less than 3 decimal places, Algorithm (3.1) needs 53 iterations while Algorithm 3.3.1 takes just 15 iterations to obtain the same goal. This observation shows that the convergence speed of Algorithm 3.3.1 is faster than Algorithm (3.1).

Number of iteration.	Algorithm (3.1)	Algorithm 3.3.1
1	17.801304917618964	17.826809506966804
3	18.238459918161315	17.896347714673194
5	18.998465909761100	11.038168038343564
7	18.550509556843764	4.104266737615634
9	17.820863132624478	2.019351546352653
11	16.751374190962910	2.681677869669982
13	14.336675808934617	1.926464479485708
15	11.487166440018544	7.521568964188269e-04
17	9.510101400429985	0
19	6.199317809338941	0
21	4.524962881620460	0
23	2.637915932224955	0
25	1.302567649926143	0
27	1.653802630048002	0
29	1.820947035691583	0
31	1.819006839555166	0
37	0.892617910214436	0
39	0.598832844725429	0
41	0.405996149023377	0
43	0.268613850479600	0
45	0.161184227430202	0
50	0.012670530567700	0
52	0.002881426384463	0
53	0.001361524082377	0
54	6.428424686431453e-04	0
55	0	0
56	0	0
58	0	0
60	0	0
70	0	0

TABLE 3.1: The evaluation of $\|x_n\|_2$ as number of iteration increases for Algorithm (3.1) and Algorithm 3.3.1

3.4 Applications

3.4.1 Convex concave saddle point problem

Consider $\mathcal{H}_1$ and $\mathcal{H}_2$ are two Hilbert spaces. To define the saddle point problem, we consider the following convex functions:

- $f : \mathcal{H}_1 \to \mathbb{R}_\infty$.
- $g : \mathcal{H}_1 \to \mathbb{R}_\infty$ is differentiable with L_g-Lipschitz gradient.
- $h^* : \mathcal{H}_2 \to \mathbb{R}_\infty$.
- $k^* : \mathcal{H}_2 \to \mathbb{R}_\infty$ is differentiable with L_k-Lipschitz gradient.

The saddle point problem is defined as follows:

$$\min_{x\in\mathcal{H}_1} \max_{y\in\mathcal{H}_2} f(x) + g(x) + \langle Lx, y\rangle - h^*(y) - k^*(y), \tag{3.17}$$

where $L : \mathcal{H}_1 \to \mathcal{H}_2$ is a linear bounded operator. Let $\mathbb{S}$ denote the solution set of saddle point problem (3.17).

Define operators A and B on $\mathcal{H}_1 \times \mathcal{H}_2$ by

$$A := \begin{bmatrix} \partial f & L^* \\ -L & \partial h^* \end{bmatrix} \text{ and } B := \begin{bmatrix} \nabla g & 0 \\ 0 & \nabla k^* \end{bmatrix}.$$

Note that A and B are maximally monotone operators. Thus using the above argument and KKT conditions, saddle point problem (3.17) can be formulated as the following inclusion problem

$$0 \in (A+B)\begin{bmatrix} x \\ y \end{bmatrix}.$$

We show that saddle point problem can be solved by adapting our Algorithm 3.3.1. For this, we consider the linear operator

$$M = \begin{bmatrix} L_g Id & 0 \\ 0 & L_k Id \end{bmatrix}.$$

The convergence analysis can be summarized in the following theorem.

Theorem 3.4.1. *Suppose that solution set* $\mathbb{S}$ *is nonempty. Let* $\lambda \in (0, 1]$ *and let the parameters* $\{\alpha_n\}$ *and* $\{\beta_n\}$ *satisfy Assumption 3.3.1. Let* $\{(x_n, y_n)\}$ *be the sequence in* $\mathcal{H}_1 \times \mathcal{H}_2$ *generated from initial points* $(x_0, y_0) = (x_1, y_1) \in \mathcal{H}_1 \times \mathcal{H}_2$ *and defined by*

$$\begin{cases} \mu_n = x_n + \alpha_n(x_n - x_{n-1}) \\ \nu_n = y_n + \alpha_n(y_n - y_{n-1}) \\ u_n = (1-\beta_n)\mu_n + \beta_n \xi^{-1}\{\chi_1(\mu_n) - \chi_2(\nu_n)\} \\ v_n = (1-\beta_n)\mu_n + \beta_n \xi^{-1}\{\zeta_1(\mu_n) + \zeta_2(\nu_n)\} \\ x_{n+1} = \xi^{-1}\{\chi_1(u_n) - \chi_2(v_n)\} \\ y_{n+1} = \xi^{-1}\{\zeta_1(u_n) + \zeta_2(v_n)\}, n \in \mathbb{N}, \end{cases} \tag{3.18}$$

where $\chi_1 \equiv (L_g Id - \lambda\nabla g)(L_k Id + \lambda\partial h^*)$, $\chi_2 \equiv \lambda L^*(L_k Id - \lambda\nabla k^*)$, $\zeta_1 \equiv \lambda L(L_g Id - \lambda\nabla g)$, $\zeta_2 \equiv (\lambda\partial f + L_g Id)(L_k Id - \lambda\nabla k^*)$ *and* $\xi \equiv (\lambda\partial f + L_g Id)(L_k Id + \lambda\partial h^*) + \lambda^2 LL^*$. *Then sequence* $\{(x_n, y_n)\}$ *converges weakly to a point in the solution set* $\mathbb{S}$.

Proof. Since g is convex with L_g-Lipschitz continuous gradient, it follows from Baillon-Hadded Theorem [9] that ∇g is cocoercive with respect to L_g^{-1}. Similarly,

∇k^* is cocoercive with respect to ${L_k}^{-1}$. For $(x,y),(\xi,\zeta) \in \mathcal{H}_1 \times \mathcal{H}_2$, we have

$$\begin{aligned} & \langle B(x,y) - B(\xi,\zeta), (x,y) - (\xi,\zeta)\rangle_{\mathcal{H}_1\times\mathcal{H}_2} \\ = \; & \langle \nabla g(x) - \nabla g(\xi), x - \xi\rangle_{\mathcal{H}_1} + \langle \nabla k^*(y) - \nabla k^*(\zeta), y - \zeta\rangle_{\mathcal{H}_2} \\ \geq \; & L_g^{-1}\|\nabla g(x) - \nabla g(\xi)\|^2_{\mathcal{H}_1} + L_k^{-1}\|\nabla k^*(y) - \nabla k^*(\zeta)\|^2_{\mathcal{H}_2}. \end{aligned}$$

Thus, B is M-cocoercive. With the above choice of A, B and M, Algorithm 3.3.1 reduces to Algorithm (3.18). Since parameters $\{\alpha_n\}$ and $\{\beta_n\}$ satisfy conditions $(B1)$, $(B2)$ and $(B3)$, Algorithm (3.18) converges weakly to a point in the solution set $\mathbb{S}$. □

Remark 3.4.1. *In order to solve saddle point problem (3.17), when B is M-cocoercive, the Algorithm (3.1) proposed by Lorenz and Pock [61] can be written as follows:*

$$\begin{cases} \mu_n = x_n + \alpha_n(x_n - x_{n-1}) \\ \nu_n = y_n + \alpha_n(y_n - y_{n-1}) \\ x_{n+1} = \xi^{-1}\{\chi_1(\mu_n) - \chi_2(\nu_n)\} \\ y_{n+1} = \xi^{-1}\{\zeta_1(\mu_n) + \zeta_2(\nu_n)\}, n \in \mathbb{N}, \end{cases} \tag{3.19}$$

where operators $\xi, \chi_1, \chi_2, \zeta_1$ and ζ_2 are as in Theorem 3.4.1. If parameters α_n and λ satisfy the assumptions as in Theorem 3.1.1, then the sequence $\{(x_n, y_n)\}$ generated by Algorithm (3.19) converges weakly to a point in solution set $\mathbb{S}$.

3.4.2 Lasso problem

The Lasso problem is extensively used in the field of signal processing, image processing and machine learning (see [14, 27, 71]). Many problems arising in these fields can be expressed as Lasso problem. For the choice of $f = \rho\|x\|_1$, $g = \frac{1}{2m}\|Ax - b\|^2$ and $h^* = k^* = L = 0$, Lasso problem (2.24) can be framed as saddle point problem (3.17). Thus, we have the following result.

Corollary 3.4.1. *Suppose that the solution set of Lasso problem (2.24) is nonempty. Let the parameter $\{\alpha_n\}$, $\{\beta_n\}$ and λ satisfy Assumption 3.3.1. Consider the sequence $\{x_n\}$ generated by the following algorithm with the initial point $x_0{=}x_1$ and defined by,*

$$\begin{cases} \mu_n = x_n + \alpha_n(x_n - x_{n-1}) \\ u_n = (1-\beta_n)\mu_n + \beta_n(Id + \lambda L_g^{-1}\rho\partial\|\cdot\|_1)^{-1}\left(\mu_n - \frac{\lambda L_g^{-1}A^T(A\mu_n - b)}{m}\right) \\ x_{n+1} = (Id + \lambda L_g^{-1}\rho\partial\|\cdot\|_1)^{-1}\left(u_n - \frac{\lambda L_g^{-1}A^T(Au_n - b)}{m}\right). \end{cases} \tag{3.20}$$

Then $\{x_n\}$ converges weakly to an optimal point of Lasso problem.

Proof. Using Theorem 3.4.1, we can obtain that Algorithm (3.20) converges weakly to an optimal point of Lasso problem (2.24). □

Remark 3.4.2. *Algorithm (3.19) can be used to solve Lasso problem (2.24). For the choice of $f = \rho\|x\|_1$, $g = \frac{1}{2m}\|Ax - b\|^2$ and $h^* = k^* = L = 0$, the algorithm reduces to the following*

$$\begin{cases} \mu_n = x_n + \alpha_n(x_n - x_{n-1}) \\ x_{n+1} = (Id + \lambda L_g^{-1}\rho\partial\|\cdot\|_1)^{-1}\left(\mu_n - \frac{\lambda L_g^{-1}A^T(A\mu_n - b)}{m}\right). \end{cases} \tag{3.21}$$

With the assumptions as in Remark 3.4.1, the sequences $\{x_n\}$ generated by Algorithm (3.21) converges weakly to a solution of the Lasso problem.

3.5 Numerical Experiments

In this section, we perform numerical experiments to demonstrate the realworld applicability of the proposed algorithm. All the numerical experiments are performed in the MATLAB 2018a environment on Intel(R)core(TM)i5 processor with 8GB RAM and 64-bit operating system.

3.5.1 Regression problems

In this subsection, we compare the performance of Algorithms (3.18) and (3.19) for a regression problem on high dimensional datasets. The objective function we consider is the loss function with l_1-regularization, i.e., Lasso problem (2.24). We employ both the Algorithms (3.18) and (3.19) to solve the Lasso problem (2.24) and compare their performance on the basis of their convergence speed and accuracy. For our experiment, we consider the Lasso problem with data $(A_i, b_i), i = 1, 2, ..., m$, where $A_i = (A_{i1}, A_{i2}, ..., A_{id})^T$ are predictor variables and b_i are responses. The description of the datasets[1] is summarized in Table 3.2. Here, the total number of

Datasets	$\|V\|$	$\|E\|$	$\langle D\rangle$	$\langle K\rangle$	$\langle C\rangle$
Dolphin	62	159	3.302	5.129	0.258
Football	115	613	2.486	10.660	0.403
Jazz	198	2742	2.235	27.697	0.620
Celegansneural	297	2148	2.447	14.465	0.308
Usair97	332	2126	2.738	12.807	0.749
Netscience (NS)	379	914	6.042	4.823	0.798
Political blogs (PB)	1222	16714	2.738	27.355	0.360

TABLE 3.2: Topological information of real-world network datasets

vertices and edges in a network is represented by the symbols $|V|$ and $|E|$, respectively. $\langle D\rangle$ represents the average shortest path length, $\langle K\rangle$, the average degree, and

[1] http://www-personal.umich.edu/ mejn/netdata/

$\langle C \rangle$, the average clustering coefficient of the network. For experimental purpose, we select the inertial parameter $\alpha_n = \frac{n-1}{14n+2.5}$ and $\beta_n = 0.5 + \frac{1}{200n}$ which satisfy the conditions $(B1)$, $(B2)$ and $(B3)$. A bias column is added to dataset and we run algorithms for maximum 1000 iteration. We select $\alpha \times \|Ab^T\|_\infty$ as a regularization parameter and vary α between 10^{-10} to 10^{-3} in the multiple of 0.1. The best results are shown here.

In the first experiment, we compare the performances of Algorithm (3.18) and Algorithm (3.19) on the basis of their convergence speed. We compute the difference between objective function value $F(x)$ and optimized value $F(x^*)$ at each iteration for both Algorithms (3.18) and (3.19). We initialize the experiment with point $x_0 = x_1 = 0 \in \mathbb{R}^d$. The numerical results are reported in Figure 3.8 for 1000 iterations.

In Figure 3.8, we plot the graph between $F(x) - F(x^*)$ and the number of iterations. From Figure 3.8, we can observe that Algorithm (3.18) has better convergence speed than Algorithm (3.19) for all datasets.

In the second experiment, we compare both the Algorithms (3.18) and (3.19) on the basis of their accuracy. We calculate the root mean square error (RMSE) of Algorithms (3.18) and (3.19) at each iteration. We take the initial points $x_0 = x_1 = 0 \in \mathbb{R}^d$ and plot the graph between RMSE and the number of iterations, which is shown in Figure 3.15.

From Figure 3.15, we can observe that at each iteration the RMSE value of Algorithm (3.18) is less than RMSE value of Algorithm (3.19) for all datasets. Thus, Algorithm (3.19) is more accurate than Algorithm (3.18).

Remark 3.5.1. *From experiments, we observe that Algorithm (3.18) not only have higher convergence speed but it also gives more accurate results than Algorithm (3.19)*

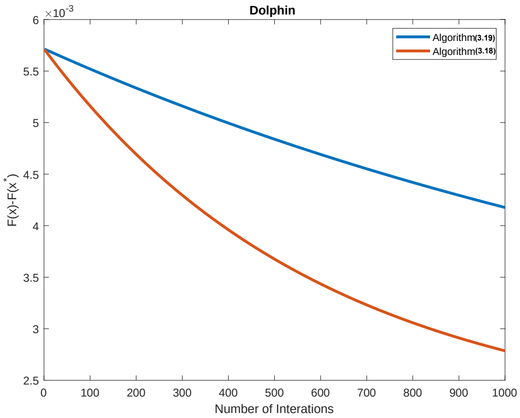

FIGURE 3.2: Dolphin.

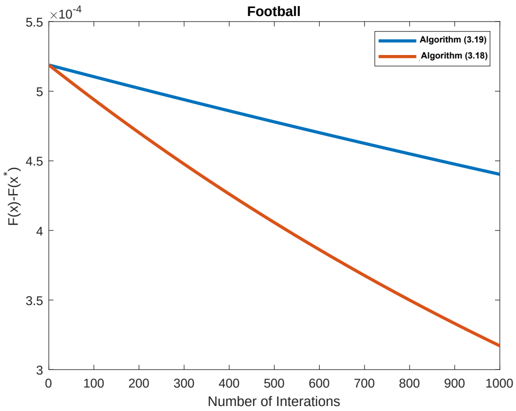

FIGURE 3.3: Football.

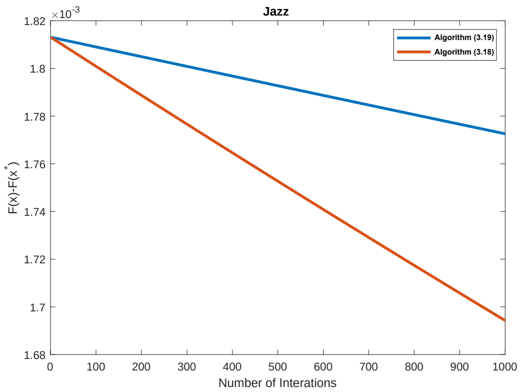

FIGURE 3.4: Jazz.

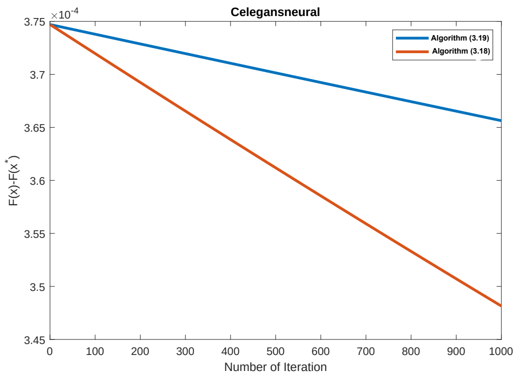

FIGURE 3.5: Celegansneural

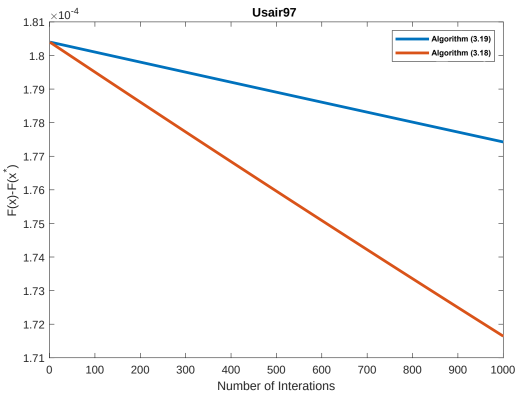

FIGURE 3.6: Usair97

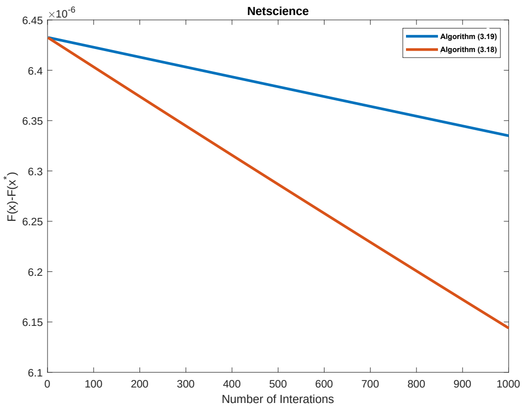

FIGURE 3.7: Netscience.

FIGURE 3.8: Value of $F(x_n) - F(x^*)$ for 1000 iterations with different datasets.

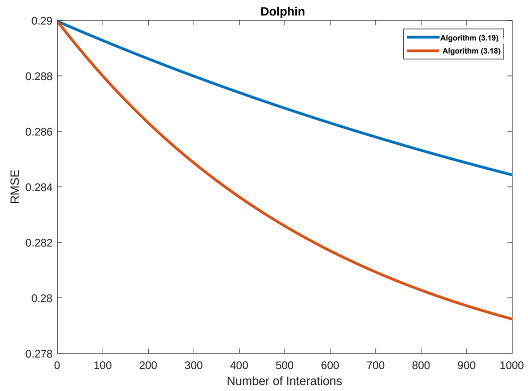

FIGURE 3.9: Dolphin.

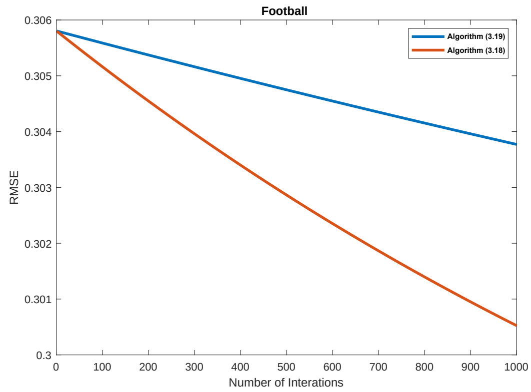

FIGURE 3.10: Football.

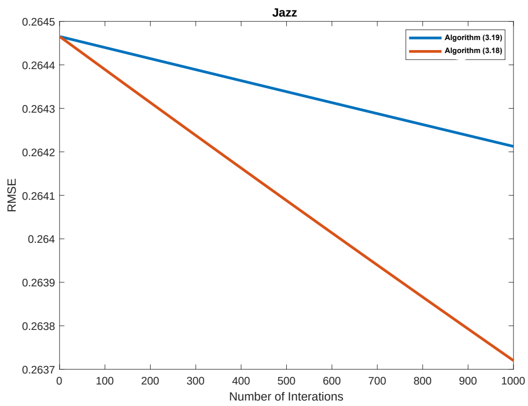

FIGURE 3.11: Jazz.

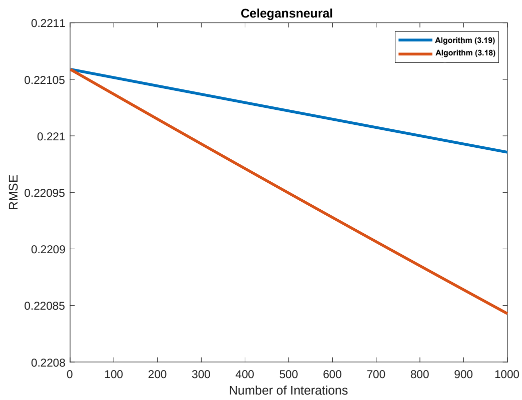

FIGURE 3.12: Celegansneural

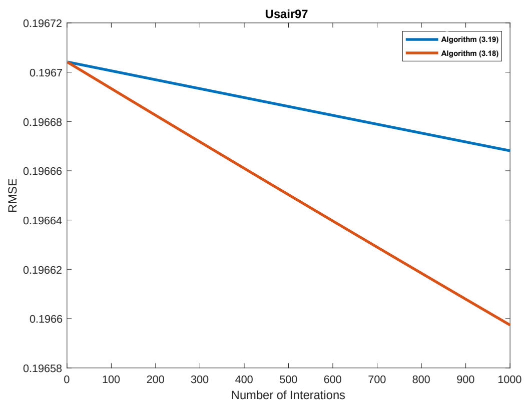

FIGURE 3.13: Usair97

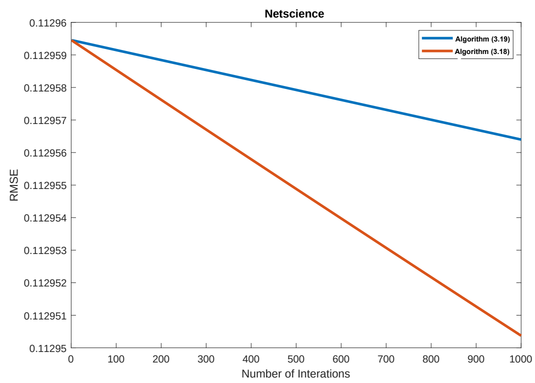

FIGURE 3.14: Netscience.

FIGURE 3.15: Behavior of root mean square error (RMSE) for different datasets.

for high dimensional datasets also. Thus, we observe that Algorithm (3.18) is equally important over Algorithm (3.19) for high dimensional datasets also, as we have obtained in numerical example 3.3.2.

3.5.2 Link prediction problems

To further analyze the proposed algorithm, we depict the practical application of the proposed Algorithm (3.18) to solve a link prediction problem. The Algorithm (3.18) is applied to predict missing links in networks (popularly known as link prediction [54, 57]). The link prediction is considered as the binary classification problem where the two classes are the link existence and link absence between two nodes. Logistic model [48] is used to classify the different links, which can be formulated as convex minimization problem, given by

$$\min_{\Theta \in \mathbb{R}^n} -\frac{1}{m}\left[\sum_{i=1}^{m} b^i log\ h_\Theta(x^i) + (1-b^i) log(1-h_\Theta(x^i))\right] + \rho\|\Theta\|_1, \tag{3.22}$$

where $h_\Theta(u) = (1 + \exp -\Theta^T u)^{-1}$ is a sigmoid function, ρ is a regularization parameter, m is the total number of node pairs, x^i's are feature vectors and b^i's indicate the existence of link between nodes. Minimization problem (3.22) reduces to saddle point problem (3.17) by assuming $f = \rho\|\Theta\|_1$, $g = \frac{1}{m}[\sum_{i=1}^{m} b^i log\ h_\Theta(x^i) + (1 - b^i)\ log(1 - h_\Theta(x^i))]$ and $h^* = k^* = L = 0$. The experiment of the link prediction is carried out in two phases, viz., feature extraction phase and regression phase. Features are automatically extracted using the autoencoder framework of deep learning [52] with two hidden layers. Each node of the network is represented using 16 features. Once the node features are extracted, edge features are computed using the binary operator. Further, the regression is applied for the best estimation of the

Datasets	Accuracy	Logistic Error
Dolphin	0.915918	0.084082
Football	0.906484	0.093516
Celegansneural	0.951133	0.048867
Usair97	0.961307	0.038693
Political blogs	0.734417	0.265583

TABLE 3.3: Result

decision boundary and accuracy is computed based on this decision boundary. The experiment is carried out on real network datasets tabulated in Table 3.2.

Accuracy. Accuracy and logistic error corresponding to seven realworld network datasets are shown in Table 3.3. The proposed method shows errors of less than 10% on all datasets except the political blogs where the error reaches up to a higher level of 26.5%. It shows the best accuracy result on coauthorship data (Netscience) compared to others.

We also compare the accuracy results of the proposed methods with some well known existing approaches (viz., common neighbors (CN) [57], Adamic/Adar (AA) [2], Resource allocation (RA) [75], Preferential attachment (PA) [7], and CAR [28]. These results are tabulated in Table 3.4, where the best value against each dataset is shown in bold-face. From Table 3.4, we observe that the proposed method shows best results on Dolphin, Football, Jazz, Celegansneural, and Usair97 datasets with significant margins. CAR is the best performing method on Netscience and Political blogs datasets. One thing to note that the accuracy of all the methods in the table is almost similar on Netscience except the PA.

	Accuracy					
Datasets	Algorithm (3.18)	CN	AA	RA	PA	CAR
Dolphin	**0.915918**	0.767566	0759134	0.773187	0.603709	0.682928
Football	**0.906484**	0.658252	0.658417	0.663202	0.526189	0.846447
Jazz	**0.930058**	0.688999	0.698540	0.719959	0.618850	0.861920
Celegansneural	**0.951133**	0.539677	0.715186	0.752523	0.684004	0.842770
Usair97	**0.961307**	0.660190	0.782063	0.820363	0.780785	0.939876
Netscience	0.986668	0.996769	0.996824	0.996742	0.719727	**0.999697**
Political blogs	0.734417	0.765682	0.777344	0.853749	0.816781	**0.964388**

TABLE 3.4: Result Comparison

3.6 Conclusion

In this chapter, we have proposed a preconditioned forward-backward algorithm to solve the monotone inclusion problem and studied its convergence behavior. The proposed algorithm is applied to solve saddle point problem. We have conducted numerical experiments to solve the regression and Link prediction problems. Numerical experiments show that the proposed algorithm has better convergence speed and accuracy than the algorithm proposed by Lorenz and Pock [61]. The proposed algorithm is also compared with some well-known existing methods to solve link prediction problems. In most of cases, the proposed algorithm outperforms the methods under consideration.

****