이학박사 학위논문

Accelerated algorithms for linearly constrained convex minimization

(선형제한조건의 수학적 최적화 문제를 위한 빠른 알고리즘들)

2014년 2월

서울대학교 대학원

수리과학부

강 명 민

Accelerated algorithms for linearly constrained convex minimization

(선형제한조건의 수학적 최적화 문제를 위한 빠른 알고리즘들)

지도교수 강 명 주

이 논문을 이학박사 학위논문으로 제출함

2013년 10월

서울대학교 대학원

수리과학부

강 명 민

강 명 민의 이학박사 학위논문을 인준함

2013년 12월

위 원 장 ______________________ (인)

부 위 원 장 ______________________ (인)

위 원 ______________________ (인)

위 원 ______________________ (인)

위 원 ______________________ (인)

Accelerated algorithms for linearly constrained convex minimization

A dissertation
submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
to the faculty of the Graduate School of
Seoul National University

by

Myeongmin Kang

Dissertation Director : Professor Myungjoo Kang

Department of Mathematical Sciences
Seoul National University

February 2014

Abstract

This dissertation introduces acceleration methods for solving the linearly constrained convex minimization problem. The proposed methods are commonly based on the extrapolation technique, which is used in accelerated proximal gradient methods proposed by Nesterov. The content of this dissertation is divided into two main algorithms. The first algorithm is the accelerated Bregman method and we numerically test accelerated Bregman method on a synthetic problem from compressive sensing and this numerical results confirm that our accelerated Bregman method is faster than the original Bregman method. The second algorithm is the inexact accelerated augmented Lagrangian method and we give the inexact stopping condition of subproblem of accelerated augmented Lagrangian method. We also develop the inexact accelerated alternating direction method of multiplier which is developed similar with inexact accelerated augmented Lagrangian method.

Key words: Augmented Lagrangian method, Bregman iteration, Compressive Sensing, Nesterov's Acceleration Method, Convex Optimization
Student Number: 2008-20274

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Let us consider the linearly constrained convex optimization problem

$$\min_x f(x) \quad \text{subject to} \quad Ax = b, \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a convex, proper and lower semi-continuous function, $A \in \mathbb{R}^{m\times n}$, and $b \in \mathbb{R}^m$. A famous application of (1.1) is the case of $f(x) = \|x\|_1$, which is called basis pursuit problem. Basis pursuit is related to compressive sensing [9] whose main concept is that a sparse signal can be recovered from incomplete information i.e. underdetermined system $Ax = b$ where $m \ll n$.

The augmented Lagrangian method (ALM) is a well-known algorithm for solving (1.1). It is one of the Lagrangian methods which allow primal and dual variables to be considered at the same time via the related constrained problem (1.1). The ALM was first proposed in [24, 41] and discussed by Rockafellar [42] as the application of the classical proximal point algorithm. This method was also studied by Bertsekas in [6]. It was also turned out to be equivalent to the Bregman method [58] proposed for solving the basis pursuit.

The alternating direction method of multipliers (ADMM) is a variant of the ALM, which often solves the following problem:

$$\min_{u,v} F(u) + G(v) \quad \text{subject to} \quad Bu + Cv = b, \tag{1.2}$$

where $F : \mathbb{R}^n \to \mathbb{R}$, $G : \mathbb{R}^p \to \mathbb{R}$ is a convex, proper and lower semicontinuous functions, $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times p}$, and $b \in \mathbb{R}^m$. The ADMM is one of the most frequently used algorithms in image processing [11, 19, 53, 54], since many unconstrained minimization problems in image processing can be converted to the constrained forms as in (1.2), using variable splitting scheme [19]. It was shown in [14, 48] that the ADMM is equivalent to the well-known algorithms, the Douglas-Rachford splitting [12] and the alternating split Bregman algorithm [19], when solving linear equality constrained optimization problems like (1.2).

Recently, researchers have been working on acceleration of the iterative algorithms. The accelerated methods rely on the previous computed iterate and two or more previously computed iterates, when computing the next iterate. First, the acceleration schemes have been developed to solve unconstrained convex minimization problems due to simplicity of the problems. For instance, an acceleration scheme was studied in [13], by using sequential subspace optimization technique and minimizing a function over an affine subspace spanned by two or more previous iterates and current gradient. The authors showed that their algorithm is faster than the iterative shrinkage thresholding algorithm (ISTA) [20] which is a well-known algorithm for solving unconstrained minimization problems. Moreover, a fast iterative shrinkage thresholding algorithm (FISTA) [4] is also proposed as an acceleration of the ISTA, based on the Nesterov's acceleration schemes [37]. More recently, accelerated schemes have been also introduced for linearly constrained convex minimization problems as (1.1) by solving its dual problem. Using the Nesterov's technique, the accelerated linearized Bregman method (ALB) [26] was developed as an acceleration of the linearized Bregman method [8]. Lastly, to solve the constrained problem (1.2), Goldstein et al [18] proposed acceleration versions of alternating algorithms such as the ADMM or alternating minimization algorithm (AMA). Very recently, He et al. [22] developed the accelerated ALM for the linearly constraints minimization problem whose objective function is differentiable. Since the problem (1.1) has only linear constraint, in this dissertation, we extend the algorithm in [22] to solve the linearly constrained minimization problem in which the object function is convex and continuous but not differentiable. By using the equivalence between the Bregman method and the ALM, and using the generalization of the accelerated ALM [22], we propose the accelerated Bregman method (ABM) for solving the linearly constrained convex minimization problem (1.1).

The algorithms for solving constrained minimization problems (1.1) or (1.2) have the common idea to derive iterative algorithms in which each iteration consists of a subproblem. In general, there are many cases where the subproblems cannot be solved exactly. Hence, inexact algorithms have been developed with analysis about inexact solutions of the subproblems. The ALM and ADMM also have subproblems, so their inexact versions have been introduced. In [21], an inexact stopping condition was provided with an appropriate upper bound of difference between exact solution and inexact solution. NG et al [38] developed an inexact version of the ADMM, whose subproblems additionally have quadratic proximal terms. Moreover, a property of the ALM related to inexact solution of subproblem was discussed in [57], when the objective function $f(x)$ is the ℓ_1-norm; a high accurate solution can be found by a few inexact subproblem steps. On the other hand, the work [5] provided a stopping condition of the subproblem in Bregman iteration using the ε-subdifferential. Recently, several researchers have worked on the inexact versions of the accelerated schemes such as the FISTA and accelerated proximal point method (APPM) [1], to solve unconstrained convex minimization problems. Villa et al [51] provided an inexact stopping criteria of the proximal operator in the FISTA, and proved that the convergence rate of the inexact FISTA is the same as that of the FISTA. And the inexact APPM was proposed in [23] with the same convergence rate with the APPM. These algorithms in [23, 51] are inexact accelerated algorithms proposed for solving unconstrained problem. None of studies have provided an inexact accelerated algorithm to solve a linearly constrained convex minimization problem so far. In this article, we propose inexact versions of accelerated ALM and accelerated ADMM [18] to solve the constrained problems (1.1) and (1.2) respectively. We introduce inexact stopping conditions of the subproblems, and prove that the convergence rate remains $\mathcal{O}(\frac{1}{k^2})$ which is a common convergence rate of accelerated schemes using Nesterov's extrapolation technique.

The remainder of this dissertation is organized as follows. Chapter 2 introduces the well-known algorithms for solving linearly constrained convex optimization and accelerated schemes for solving unconstrained convex optimization. Main contributions of this dissertation are introduces in chapter 3.

At first, we describe the accelerated ALM, propose our ABM and analyze the convergence result of this algorithm. In next, we describe the inexact accelerated ALM and the inexact accelerated ADMM. We give the stopping criteria for any subproblem solvers. Lastly, we provide the numerical tests for all proposed algorithms. For ABM, we solve the linearly constrained ℓ_1 and generalized ℓ_1 minimization problem and compare the performance of our ABM with that of original Bregman method. For the inexact accelerated ALM, we numerically test on linearly constrained ℓ_1-ℓ_2 minimization in two parts. In first part, we confirm the convergence of the inexact accelerated ALM as use various subproblem solvers. In second part, we compare the performance of our proposed algorithm with that of state-of-the-art algorithms. For inexact accelerated ADMM, we proposed the new model for multiplicative image denoising and we compare the performance of our proposed algorithm for our model with that of alternating minimization algorithm for TV model. In chapter 4, we give the conclusion of our dissertation and discuss the future works.

Chapter 2

Previous Methods

In this chapter, we introduce the previous algorithms for solving convex optimization problems. At first, we explain augmented Lagrangian method(ALM), Bregman methods and alternating direction method of multiplier(ADMM) for solving the problem (1.1) or (1.2). We consider the relations of these algorithms. We also introduce accelerating algorithms for solving unconstrained minimization problem.

2.1 Mathematical Preliminary

In this section, we introduce the mathematical concepts and notations which will be used in our work. Let us consider a convex function $f : \mathbb{R}^n \to \mathbb{R}$ and a vector $x \in \mathbb{R}^n$. The subdifferential $\partial f(x)$ of f at x is defined by

$$\partial f(x) = \{s : f(y) - f(x) \geq s^T(y - x) \text{ for all } y \in \mathbb{R}^n\}.$$

There are several properties of subdifferential. If $f : \mathbb{R}^n \to \mathbb{R}$ is a proper, convex and lower semicontinuous, then the subdifferential of f at any $x \in \text{dom}(f)$ exists. Moreover, if f is differentiable, the subdifferential of f at x has only one element and satisfies $\partial f(x) = \nabla f(x)$.

The subdifferential can be used to solve the the unconstrained minimization problem, which is given by the following identity:

$$f(x) = \min_{y} f(y) \Leftrightarrow 0 \in \partial f(x). \tag{2.1}$$

The subdifferential ∂f also satisfies the following relation:

$$x^* \in \partial f(x) \Leftrightarrow x \in \partial f^*(x), \tag{2.2}$$

where $f^* : \mathbb{R}^n \to \mathbb{R}$ is the conjugate function of f defined as $f^*(p) = \sup_x\{\langle x, p\rangle - f(x)\}$. In addition, it can be easily verified that the subdifferential of f is a monotone operator, i.e. $\langle p-q, x-y\rangle \geq 0$, where $p \in \partial f(x)$ and $q \in \partial f(y)$. We define the ϵ-subdifferential of f at x by the set

$$\partial_\epsilon f(x) = \{s | f(y) - f(x) \geq s^T(y-x) - \epsilon \text{ for all } y \in \mathbb{R}^n\},$$

where ϵ is a positive number. Similar with properties of subdifferential, it holds

$$0 \in \partial_\epsilon f(x) \Leftrightarrow f(x) \leq \inf f + \epsilon,$$
$$x^* \in \partial_\epsilon f(x) \Leftrightarrow x \in \partial_\epsilon f^*(x^*).$$

The proximal point of y with respect to λf is defined as follows,

$$\text{prox}_{\lambda f}(y) := \arg\min_x \left\{ f(x) + \frac{1}{2\lambda}\|x-y\|_2^2 \right\}$$

and the mapping $\text{prox}_{\lambda f}$ is called proximity operator of λf, which was introduced in Moreau [31]. Let $\Phi_\lambda(x) = F(x) + \frac{1}{2\lambda}\|x-y\|_2^2$. By the first order optimality condition for unconstrained minimization problem (2.1), we have following equivalent identities:

$$x = \text{prox}_{\lambda f}(y) \Leftrightarrow 0 \in \partial\Phi_\lambda(x) \Leftrightarrow \frac{y-x}{\lambda} \in \partial f(x).$$

The last term yields

$$\text{prox}_{\lambda f}(y) = (I + \lambda\partial f)^{-1}(y).$$

The convex function f is called a strongly convex function with the modulus σ_f, if and only if there exist a constant $\sigma_f > 0$ such that the function $f(x) -$

$\frac{\sigma}{2}\|x\|_2^2$ is convex. If f is a strongly convex function with the modulus σ_f, then the following inequality is satisfied for every x and y:

$$\langle p - q, x - y \rangle \geq \sigma_f \|x - y\|_2^2,$$

where $p \in \partial f(x)$ and $q \in \partial f(y)$. We also note an important property of strongly convex function related to its conjugate function: If f is a strongly convex with σ_f, the conjugate function f^* of f is differentiable and its gradient function ∇f^* is Lipschitz continuous function with the Lipschitz constant $L(\nabla f^*) = \sigma_f^{-1}$.

Now we consider the dual problem of linearly constrained minimization problem (1.1). First, the Lagrangian function for the problem (1.1) is defined as

$$\mathcal{L}(x, \lambda) = f(x) - \lambda^T (Ax - b),$$

where λ is called Lagrangian multiplier vector or dual variable. Then the Lagrangian dual function for (1.1) is given by

$$D(\lambda) = \inf_x \mathcal{L}(x, \lambda) = \inf_x (f(x) - \lambda^T (Ax - b)),$$

i.e. it has the minimum value of the Lagrangian function over x. This can be also represented as

$$D(\lambda) = -f^*(A^T \lambda) + \lambda^T b.$$

Therefore, the dual problem of (1.1) is defined as maximize the Lagrangian dual function over dual variable λ:

$$\max_\lambda D(\lambda) = \max_\lambda (-f^*(A^T \lambda) + \lambda^T b). \tag{2.3}$$

The original problem (1.1) is called primal problem.

Lastly, we briefly describe the duality that refers to a relation of dual problem and primal problem. Let d^* and p^* be the optimal values of the dual (2.3) and primal (1.1) problems, respectively. In the case of linear constraints, the strong duality is satisfied, i.e., the optimal duality gap is zero; $d^* = p^*$. And we say that (x, λ) for some $x \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$ satisfies KKT (Karush-Kuhn-Tucker) optimality conditions [25] of the problem (1.1) if (x, λ) satisfied the following conditions:

$$\partial f(x) - A^T \lambda = 0, \text{ and } Ax = b.$$

The KKT condition is the necessary and sufficient optimal condition for primal problem (1.1) and its dual (2.3). That is, x^* and λ^* are any primal and dual optimal points with zero duality gap if and only if they satisfy the KKT conditions.

2.2 The algorithms for solving the linearly constrained convex minimization

We introduce some algorithms which can be applied the linearly constrained convex minimization problem (1.1) or (1.2). We explain details of the augmented Lagrangian method which was briefly introduced in previous chapter. We also explain the Bregman method which was proposed in [58] at first for solving the basis pursuit. We observe that the equivalence between the Bregman method and the augmented Lagrangian method. There are some weaknesses of Bregman method for solving the basis pursuit, so, its variant algorithm was developed in [8], it is called linearized Bregman method. The linearized Bregman method is equivalent to a gradient descent method, so, based on this fact, the accelerated linearized Bregman method was developed in [26]. Finally, we explain the alternating direction method of multipliers which is a variant of augmented Lagrangian method and can be solves the problem (1.2).

2.2.1 Augmented Lagrangian Method

In this section, we consider the problem (1.1) and introduce the previous algorithms for solving the problem (1.1).

The augmented Lagrangian function for the problem (1.1) is defined as follows:

$$\mathcal{L}_A(x, \lambda, \tau) = f(x) - \lambda^T(Ax - b) + \frac{\tau}{2}\|Ax - b\|_2^2,$$

where λ is a Lagrange multiplier vector and $\tau > 0$ is a parameter. The augmented Lagrangian method(ALM) minimizes the augmented Lagrangian function with respect to x for fixed Lagrange multiplier λ_k, after then, update the λ_k.

Algorithm 1 Augmented Lagrangian Method(ALM)

1: **Initialization** : Choose $\tau > 0$ and $\lambda_0 = \mathbf{0}$.

2: **repeat**

3: $\quad x_{k+1} = \arg\min_x f(x) - (\lambda_k)^T(Ax - b) + \frac{\tau}{2}\|Ax - b\|_2^2,$

4: $\quad \lambda_{k+1} = \lambda_k - \tau(Ax_{k+1} - b)$

5: **until** *a stopping criterion is satisfied.*

By Fermat's rule of the third step in Algorithm 1, we get

$$\mathbf{0} \in \partial f(x_{k+1}) - A^T\lambda_k + \tau A^T(Ax_{k+1} - b),$$

i.e. λ_{k+1} in the fourth step is a subgradient of f at x_{k+1}. The ALM was proposed in [24] and [41] and Rockafellar at al.[42] and Bertsekas [6] were also discussed. The augmented Lagrangian method is equivalent to the Bregman method [58] that is well-known method for basis pursuit. We explain the further details of the Bregman method in subsection 2.2.2.

The accelerated augmented Lagrangian method (AALM) was proposed in [22] for the linearly constrained minimization whose objective function is differentiable and x is included in some closed convex set $\mathcal{X}$. They proved that the convergence rate of the AALM is $\mathcal{O}(\frac{1}{k^2})$ for each iteration k, while the convergence rate of the ALM is $\mathcal{O}(\frac{1}{k})$.

In general, the third step in AALM cannot solve exactly, i.e. that does not have the closed form solution. For example, it happens when applying (accelerated) ALM to the compressive sensing related to basis pursuit ($f(x) = \|x\|_1$). Since the convergence result of AALM was proved under the assuming that the subproblem solves exactly, that convergence analysis cannot be applied in many application. Thus, we develop the inexact accelerated augmented Lagrangian method (I-AALM) in next chapter.

2.2.2 Bregman Methods

In this subsection, we introduce the Bregman method and linearized Bregman method. The Bregman method [39] was proposed for solving total variation-based image restoration. The Bregman distance with respect to a convex, lower semicontinuous, proper function $f(\cdot)$ with points u and v is defined as

$$D_f^p(u,v) = f(u) - f(v) - p^T(u-v),$$

where p is an element in $\partial f(v)$, i.e., a subdifferential of f at v. The Bregman method for solving (1.1) can be expressed as Algorithm 2.

Algorithm 2 Original Bregman Method

1: **Initialization** : $\gamma > 0, x^0 = \mathbf{0}$ and $p^0 = 0$
2: **repeat**
3: $\quad x_{k+1} = \arg\min_x \quad D_f^{p_k}(x, x_k) + \frac{\gamma}{2}\|Ax - b\|_2^2,$
4: $\quad p_{k+1} = p_k - \gamma A^T(Ax_{k+1} - b)$
5: **until** *a stopping criterion is satisfied.*

By Fermat's rule [44, Theorem 10.1] and the fourth step in original Bregman method, we have

$$\mathbf{0} \in \partial f(x_{k+1}) - p_k + \gamma A^T(Ax_{k+1} - b),$$

Hence p_{k+1} in the second step in original Bregman method is a subgradient of f at x_{k+1}. Note that the original Bregman method does not have the parameter γ (i.e., γ is set as 1). Instead the scaling parameter μ is used in object function, for example, $f(x) = \mu(\|x\|_1 + \frac{\beta}{2}\|x\|_2^2)$. We can get the same solution by setting $\gamma = \frac{1}{\mu}$.

Now, we slightly modify the origianl Bregman method as follows:

$$\begin{cases} x_{k+1} = \arg\min_x f(x) + \frac{\gamma}{2}\|Ax - b_k\|_2^2 \\ b_{k+1} = b_k - (Ax_{k+1} - b) \end{cases} \tag{2.4}$$

starting with $x_0 = \mathbf{0}$ and $b_0 = b$. The following lemma gives the condition of the equivalence between the updating step of x_{k+1} in the original Bregman method and that in the modified Bregman method (2.4).

Lemma 2.2.1. *x_{k+1} computed by original Bregman method equals x_{k+1} computed by (2.4) if and only if*

$$p_k = \gamma A^T(b_k - b). \tag{2.5}$$

Proof. It is obvious by just comparing x_{k+1} in original Bregman method and x_{k+1} in (2.4). □

By using Lemma 2.2.1 and mathematical induction, we can establish the following theorem.

Theorem 2.2.1. *The modified Bregman method (2.4) is equivalent to the original Bregman method.*

It was proved in [58] that the Bregman method is equivalent to the augmented Lagrangian method which was introduced in section 2.2.1. For completeness, we provide a proof of the equivalence between the Bregman method and the augmented Lagrangian method. By comparing x_{k+1} in (2.4) and x_{k+1} computed by the augmented Lagrangian method, we obtain the following technical lemma to prove the equivalence. The proof is simple, so we omit it.

Lemma 2.2.2. *The x_{k+1} of the first step in (2.4) is equal to that of the first step of the augmented Lagrangian method with $\tau = \gamma$ if and only if*

$$b_k = b + \frac{\lambda_k}{\gamma}. \tag{2.6}$$

Theorem 2.2.2. *The original Bregman method is equivalent to the augmented Lagrangian method starting with $\tau = \gamma$, $\lambda_0 = \mathbf{0}$.*

Proof. We show that (2.6) holds for all integers $k \geq 0$ using induction. If $k = 0$, (2.6) holds by the initial conditions $b_0 = b$ and $\lambda_0 = \mathbf{0}$. Suppose that the (2.6) holds for k. The x_{k+1} in (2.4) equals the x_{k+1} in the augmented Lagrangian method according to Lemma 2.2.2. Thus, we have

$$\begin{aligned} b_{k+1} &= b_k - (Ax_{k+1} - b) \\ &= b + \frac{\lambda_k}{\gamma} - (Ax_{k+1} - b) \\ &= b + \frac{\lambda_k}{\gamma} + \frac{\lambda_{k+1} - \lambda_k}{\gamma} \\ &= b + \frac{\lambda_{k+1}}{\gamma}, \end{aligned}$$

where the first equality is from b_{k+1} in (2.4) and the third equality is from updating step of λ_{k+1} in the augmented Lagrangian method. Thus, the Bregman method is equivalent to the augmented Lagrangian method according to Theorem 2.2.1 and Lemma 2.2.2. □

In general, the subproblem in the first step of original Bregman method does not have the closed form solution. Hence, we have to use other iterative methods to solve the subproblem of the Bregman method. This often takes times to solve the subproblem. In order to solve this difficulty, the linearized Bregman method [8] replaces $\|Ax-b\|_2^2$ with its linearization term $(A^T(Ax_k-b))^T x$ and adds the proximal term to that replacement.

Algorithm 3 Linearized Bregman Method

1: **Input** : $\delta > 0$ and $p_0 = \mathbf{0}$.
2: **repeat**
3: $\quad x_{k+1} = \arg\min_x \quad D_f^{p_k}(x, x_k) + (A^T(Ax_k - b))^T x + \dfrac{1}{2\delta}\|x - x_k\|_2^2,$
4: $\quad p_{k+1} = p_k - A^T(Ax_k - b) - \frac{1}{\delta}(x_{k+1} - x_k).$
5: **until** *a stopping criterion is satisfied.*

Similar to the Bregman method,

$$\mathbf{0} \in \partial f(x_{k+1}) - p_k + A^T(Ax_k - b) + \frac{1}{\delta}(x_{k+1} - x_k)$$

by the optimality condition of x_{k+1}. Hence p_{k+1} is also in the subdifferential $\partial f(x_{k+1})$.

In [8, 40], it was proved that if $0 < \delta < \frac{2}{\|AA^T\|_2}$, then x_k in the linearized Bregman method converges to the solution of

$$\min_x f(x) + \frac{1}{2\delta}\|x\|_2^2 \quad \text{subject to} \quad Ax = b. \tag{2.7}$$

Recently, an accelerated version of the linearized Bregman method was proposed in [26]. Huang et al. [26] showed that the convergence rate of the accelerated linearized Bregman method is $\mathcal{O}(\frac{1}{k^2})$, where k is the iteration count, based on the equivalence between the linearized Bregman method and the gradient descent method, and the extrapolation scheme used in Nesterov's accelerated gradient descent method [4, 34].

Algorithm 4 Accelerated Linearized Bregman Method

1: **Input** : $\delta > 0, p_0 = \mathbf{0}$ and $\theta_{-1} = 1$

2: **repeat**

3: $\quad x_{k+1} = \arg\min_x \quad D_f^{\tilde{p}_k}(x, \tilde{x}_k) + \tau(A^T(A\tilde{x}_k - b))^T x + \frac{1}{2\delta}||x - \tilde{x}_k||_2^2,$

4: $\quad p^{k+1} = \tilde{p}_k - \frac{1}{\delta}(x_{k+1} - \tilde{x}_k) - \tau A^T(A\tilde{x}_k - b),$

5: $\quad \theta_k = \frac{1}{k+2},$

6: $\quad \alpha_k = 1 + \theta_k(\theta_{k-1}^{-1} - 1),$

7: $\quad \tilde{x}_{k+1} = \alpha_k x_{k+1} + (1 - \alpha_k)x_k,$

8: $\quad \tilde{p}_{k+1} = \alpha_k p_{k+1} + (1 - \alpha_k)p_k$

9: **until** *a stopping criterion is satisfied.*

2.2.3 Alternating direction method of multipliers

The alternating direction method of multipliers (ADMM) is a variant of the augmented Lagrangian method, which solves the problem (1.2). To solve the problem (1.2), the ALM is applicable to the problem, by setting $x = [u^T \ v^T]^T, A = [B \ C]$ and $f(x) = F(u) + G(v)$. This derives the following iterative algorithm

$$\begin{aligned} (u_{k+1}, v_{k+1}) &= \arg\min_{u,v} F(u) + G(v) + \lambda_k^T(Bu + Cv - b) \\ &\quad + \frac{\tau}{2}\|Bu + Cv - b\|_2^2 \\ \lambda_{k+1} &= \lambda_k - \tau(Bu_{k+1} + Cv_{k+1} - b). \end{aligned} \tag{2.8}$$

It is not trivial to solve the minimization problem in (2.8) since there are two variables coupled in a non-separable quadratic term. The ADMM alternatively solves by minimizing one variable (u or v) with the other variable fixed and performs only one outer iteration.

The convergence of ADMM is given under mild condition in [7]. It is well-known the equivalence between ADMM and Douglas-Rachford splitting [12] applied to the dual problem of the problem (1.2) and ADMM is closely related to the split Bregman method [19]. In a recent work [18], a fast ADMM (FADMM) was proposed, based on the Nesterov's extrapolating technique [37]. It was proved in [18] that the convergence rate of the ADMM is $\mathcal{O}\left(\frac{1}{k}\right)$

Algorithm 5 Alternating Direction Method of Multipliers(ADMM)

Input : $\tau > 0$ and λ_0.
repeat
  $u_{k+1} = \arg\min_u H(u) - (\lambda_k)^T(Bu) + \frac{\tau}{2}\|Bu + Cv_k - b\|_2^2,$
  $v_{k+1} = \arg\min_u G(v) - (\lambda_k)^T(Cv) + \frac{\tau}{2}\|Bu_k + Cv - b\|_2^2,$
  $\lambda_{k+1} = \hat{\lambda}_{k+1} - \tau(Bu_{k+1} + Cv_{k+1} - b)$
until *a stopping criterion is satisfied.*

while the convergence rate of their algorithm FADMM is $\mathcal{O}\left(\frac{1}{k^2}\right)$.

Algorithm 6 Fast Alternating Direction Method of Multipliers(FADMM)

Input : $\tau > 0, t_0 = 1$ and $\hat{\lambda}_1 = \lambda_0$.
repeat
  $u_k = \arg\min_u H(u) - (\hat{\lambda}_k)^T(Bu) + \frac{\tau}{2}\|Bu + C\hat{v}_{k-1} - b\|_2^2,$
  $v_k = \arg\min_u G(v) - (\hat{\lambda}_k)^T(Cv) + \frac{\tau}{2}\|Bu_k + Cv - b\|_2^2,$
  $\lambda_k = \hat{\lambda}_k - \tau(Bu_k + Cv_k - b),$
  $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2},$
  $\hat{v}_{k+1} = v_k + \frac{t_k - 1}{t_{k+1}}(v_k - v_{k-1}),$
  $\hat{\lambda}_{k+1} = \lambda_k + \frac{t_k - 1}{t_{k+1}}(\lambda_k - \lambda_{k-1})$
until *a stopping criterion is satisfied.*

Similarly to the I-AALM, we introduce an inexact version of the FADMM, with inexact stopping conditions of the subproblems. We also prove that the convergence rate of proposed algorithm is preserved as $\mathcal{O}\left(\frac{1}{k^2}\right)$ like the FADMM.

2.3 The accelerating algorithms for unconstrained convex minimization problem

Now, we explain some algorithms which can be applied the unconstrained convex optimization problem :

$$\min_x F(x), \tag{2.9}$$

where F is a proper, convex and lower semicontinuous(l.s.c) function. This unconstrained minimization problem 2.9 is used in various image processing problem. For example, the fundamental model in image denoising is the ROF(Rudin-Osher-Fatemi) model [46] which is as follows:

$$\min_{u \in \mathrm{BV}(\Omega)} \int_\Omega |\nabla u| + \frac{\mu}{2}(u-f)^2,$$

where Ω is a bounded domain in $\mathbb{R}^n$ with Lipschitz boundary, $\mathrm{BV}(\Omega)$ is the space of the functions with bounded variation, f is an observed image, u is the recovery image and $\int_\Omega |\nabla u|$ is the isotropic or anisotropic total variation(TV) of u. The basis pursuit denoising (BPDN) is also a well-known mathematical optimization model of the form (2.9):

$$\min_{x \in \mathbb{R}^n} \|x\|_1 + \frac{\mu}{2}\|b - Ax\|_2^2,$$

where $b \in \mathbb{R}^m$ is an observation vector, $A \in \mathbb{R}^{m \times n}$ is a measurement matrix with $m < n$ and $x \in \mathbb{R}^n$ is a solution vector. In comparison with constrained minimization problem, the unconstrained minimization problem (2.9) has simple setting. Hence, many accelerating schemes was developed and inexactness of their subproblems was analyzed. Most accelerating algorithms used Nesterov's extrapolation technique which is introduced in the accelerated proximal gradient methods [34] studied by Nesterov. In this section, we explain the famous inexact accelerated algorithms using Nesterov's technique for solving the problem (2.9).

2.3.1 Fast inexact iterative shrinkage thresholding algorithm

In this subsection, we consider $F(x) := f(x)+g(x)$ in (2.9) where $f : \mathbb{R}^n \to \mathbb{R}$ is a proper, convex, continuously differentiable and $g : \mathbb{R}^n \to \mathbb{R}$ is a proper, convex, l.s.c. function. The iterative shrinkage thresholding algorithm(ISTA) is a famous algorithm for solving the unconstrained convex optimization problem (2.9). For this algorithm, we assume that ∇f is a Lipschitz continuous function with Lipschitz constant L_f i.e.

$$\|\nabla f(x) - \nabla f(y)\| \leq L_f \|x - y\| \quad \text{for all } x, y.$$

By above assumption, the optimal solution x is a fixed point of the mapping $(I + \tau\partial g)^{-1}(I - \tau\nabla f)$ by following equivalence steps

$$\begin{aligned} \min_x f(x) + g(x) \quad &\Leftrightarrow \quad 0 \in \partial f(x) + \partial g(x) \\ &\Leftrightarrow \quad 0 \in (x + \tau\partial g(x)) - (x - \tau\partial f(x)) \\ &\Leftrightarrow \quad (I - \tau\partial f)x \in (I + \tau\partial g)x \\ &\Leftrightarrow \quad x = (I + \tau\partial g)^{-1}(I - \tau\nabla f)(x). \end{aligned}$$

Based on definition of the proximal mapping, we have the following equivalences:

$$\begin{aligned} & x_k && = (I + \tau\partial g)^{-1}(I - \tau\nabla f)(x_{k-1}) \\ \Leftrightarrow \quad & x_k && = \arg\min_x \tau g(x) + \frac{1}{2}\left\|x - (x_{k-1} - \tau\nabla f(x_{k-1}))\right\|_2^2, \\ \Leftrightarrow \quad & x_k && = \arg\min_x g(x) + \frac{1}{2\tau}\left\|x - (x_{k-1} - \tau\nabla f(x_{k-1}))\right\|_2^2, \\ \Leftrightarrow \quad & x_k && = \arg\min_x f(x_{k-1}) + \langle x - x_{k-1}, \nabla f(x_{k-1})\rangle + \frac{1}{2\tau}\|x - x_{k-1}\|_2^2 + g(x), \end{aligned}$$

When $\tau = \frac{1}{L_f}$, the final equality of above equations is a quadratic approximation of F at x :

$$f(y) + \langle x - y, \nabla f(y)\rangle + \frac{L_f}{2}\|x - y\|_2^2 + g(x)$$

and the ISTA for the problem (2.9) iterates the following step

$$x_{k+1} = \arg\min_y f(x_k) + \langle y - x_k, \nabla f(x_k)\rangle + \frac{L_f}{2}\|y - x_k\|_2^2 + g(y).$$

Algorithm 7 Iterative Shrinkage Thresholding Algorithm(ISTA)

Input : x_0 and L_f is the Lipschitz constant of ∇f

repeat

$$x_k = \arg\min_y f(x_{k-1}) + \langle \nabla f(x_{k-1}), y - x_{k-1} \rangle + \frac{L_f}{2}\|y - x_{k-1}\|_2^2 + g(y)$$

until *a stopping criterion is satisfied.*

When $g(x) = 0$, ISTA is same with gradient method. In smooth setting, i.e. $g \equiv 0$, the accelerated proximal gradient method with convergence rate $\mathcal{O}\left(\frac{1}{k^2}\right)$ for iteration number k was developed by Nesterov [34]. Bect et al. [4] extended this accelerated Nesterov's proximal gradient method and developed fast iterative shrinkage thresholding algorithm (FISTA) when $g(x)$ is a proper, convex, l.s.c function. They proved that the convergence rate of ISTA is $\mathcal{O}\left(\frac{1}{\epsilon}\right)$, while the convergence rate of FISTA is $\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\right)$. In the concrete, they proved the following inequities :

$$\begin{aligned} F(x_k) - F(x^*) &\leq \frac{L_f\|x_0 - x^*\|_2^2}{2k}, \\ F(x_k) - F(x^*) &\leq \frac{2L_f\|x_0 - x^*\|_2^2}{(k+1)^2}, \end{aligned}$$

where x^* is an optimal solution of the problem (2.9).

Algorithm 8 Fast ISTA(FISTA)

1: **Input** : $y_1 = x_0, t_1 = 1$ and L_f is the Lipschitz constant of ∇f

2: **repeat**

3: $\quad x_k = \arg\min_y f(y_k) + \langle \nabla f(y_k), y - y_k \rangle + \frac{L_f}{2}\|y - y_k\|_2^2 + g(y),$

4: $\quad t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2},$

5: $\quad y_{k+1} = x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1})$

6: **until** *a stopping criterion is satisfied.*

We also confirm that the FISTA is faster than ISTA by numerical test for BPDN . In this test, we set $n = 1000, m = 500$. The measurement matrix A

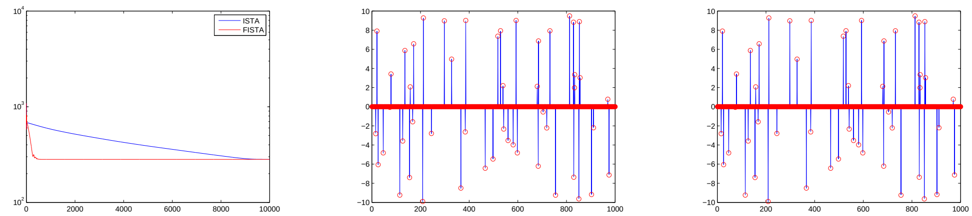

Figure 2.1: Results from BPDN. 1st column : Plot the object function value at each iteration, 2nd column : Plot the recover vector (blue) for FISTA and the solution vector x (red). 3rd column : Plot the recover vector (blue) for ISTA and the solution vector (red).

is chosen by standard Gaussian distribution $\mathcal{N}(0,1)$ and the solution vector x is a sparse vector with sparsity k(number of the nonzero elements) whose nonzero elements are randomly selected from uniform distribution on interval $(0,1)$. We add the Gaussian noise $\mathcal{N}(0,0.3)$ to the observation vector b.

In Figure 2.1, we observe that the object function value decreases as iteration increases. We also see that the decreasing speed of FISTA is faster than that of ISTA. The FISTA terminate at about $300 \sim 400$ iterations, while ISTA terminate at about $9000 \sim 10000$ iterations. Hence, we certify that FISTA is more efficient algorithm than ISTA from numerical test and theoretical result. If these algorithms iterates sufficiently large, the recovery solution is similar with original solution x in case of both two algorithms from Figure 2.1.

In [28, 51], the inexact versions of FISTA were proposed, where x_k in FISTA need not be the exact minimizer of the subproblem. We explain the stopping conditions which was introduced in [28]. We define a map $q_j : \mathbb{R}^n \to \mathbb{R}$ by

$$q_k(x) = f(y_k) + \langle \nabla f(y_k), x - y_k \rangle + \frac{L_f}{2}\|x - y_k\|_2^2.$$

Let $\{\epsilon_k\}$ and $\{\xi_k\}$ be sequences of nonnegative numbers such that their series converges, i.e. $\sum_{k=1}^{\infty} \epsilon_k < \infty, \sum_{k=1}^{\infty} \xi_k < \infty$. The inexact minimizer x_k of subproblem satisfies the stopping conditions

$$F(x_k) \leq q_k(x_k) + g(x_k) + \frac{\xi_k}{2t_k^2}, \tag{2.10}$$

$$\nabla f(y_k) + L(x_k - y_k) + \gamma_k = \delta_k \text{ with } \|\delta_k\|_2 \leq \frac{\epsilon_k}{\sqrt{2L_f}t_k}, \tag{2.11}$$

where $\gamma_k \in \partial_{\frac{\xi_k}{2t_k^2}} g(x_k)$.

Algorithm 9 Fast inexact ISTA

1: **Input** : $y_1 = x_0, t_1 = 1$ and L_f is the Lipschitz constant of ∇f

2: **repeat**

3: Find an approximate minimizer

$$x_k \approx \arg\min_y f(y_k) + \langle \nabla f(y_k), y - y_k \rangle + \frac{L_f}{2}\|y - y_k\|_2^2 + g(y),$$

satisfying the stopping conditions (2.10), (2.11).

4: $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$,

5: $y_{k+1} = x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1})$

6: **until** *a stopping criterion is satisfied.*

The authors proved in [28] that the convergence rate of this algorithm is $\mathcal{O}(\frac{1}{k^2})$ where k is number of iterations, i.e.,

$$0 \leq F(x_k) - F(x^*) \leq \frac{4}{(k+1)^2}\left((\sqrt{\tau} + \bar{\epsilon}_k)^2 + 2\bar{\xi}_k\right),$$

where $\bar{\epsilon}_k = \sum_{j=1}^{k} \epsilon_j, \bar{\xi}_k = \sum_{j=1}^{k}(\xi_j + \epsilon_j^2)$ and $\tau = \frac{L_f}{2}\|x_0 - x^*\|_2^2$.

2.3.2 Inexact accelerated proximal point method

In this subsection, we only assume that F is a proper, convex, closed, l.s.c. function. The iterative scheme of proximal point method is

$$x_{k+1} = \text{Prox}_{\lambda_k F}(x_k),$$

where $\{\lambda_k\}$ is a positive parameters and nondecreasing sequence. The proximal point algorithm was introduced by Martinet first [30] and later popularized by Rockafellar [43]. Denote that F_* is the optimal value of the problem (2.9). In [17], the sequence of object function values $F(x_k)$ for each iteration k converges to F_* under minimal assumption λ_k's and the global convergence rate of proximal point method $F(x_k) - F^* \leq \mathcal{O}(1/k)$ has been shown when F_* is attained.

Resorting to the ideas contained in Nesterov's work [35, 36], Guler et al. [16] devises an elegant way to accelerated version of the proximal point method. It was proved that the convergence rate of accelerated proximal point method satisfies $F(x_k) - F^* \leq \mathcal{O}(1/k^2)$ for each iteration k, if the minimum F^* is attained. It has been known that this convergence rate is optimal for a first order method in the sense defined in [33].

Algorithm 10 Accelerated proximal point method

1: **Input** : A feasible starting point x_0, nondecreasing and positive sequence $\{\lambda_k\}$, $\nu_0 = x_0$, and $A_0 = A > 0$.

2: **repeat**

3: Calculate

$$\alpha_k = \frac{\sqrt{(A_k\lambda_k)^2 + 4A_k\lambda_k} - A_k\lambda_k}{2}$$

4: $y_k = (1 - \alpha_k)x_k + \alpha_k\nu_k$

5: $x_{k+1} = \mathrm{Prox}_{\lambda_k F}(y_k)$

6: $\nu_{k+1} = \nu_k + \frac{1}{\alpha_k}(x_{k+1} - y_k)$

7: $A_{k+1} = (1 - \alpha_k)A_k$

8: **until** *a stopping criterion is satisfied.*

In general, very often in applications, a proximity operator does not have closed form formula. For example, there are BPDN problem and image deblurring with total variation [10]. In [47], the authors proposed accelerated inexact proximal point methods and analyzed the convergence of these algorithms. They provided two approximate conditions of proximity operator and two versions of accelerated inexact proximal point method. A type 1 approximation of $\text{prox}_{\lambda F}(y)$ with $\epsilon > 0$ is defined by

$$0 \in \partial_{\frac{\epsilon^2}{2\lambda}} \Phi_\lambda(z),$$

and is written by $z \approx_1 \text{prox}_{\lambda F}(y)$. It has important property to note that if $z \approx_1 \text{prox}_{\lambda F}(y)$ with ϵ, then

$$z \in \text{dom} F \quad \text{and} \quad \|z - \text{prox}_{\lambda F}(y)\|_2 \leq \epsilon.$$

A type 2 approximation of $\text{prox}_{\lambda F}(y)$ with $\epsilon > 0$ is defined by

$$\frac{y - z}{\lambda} \in \partial_{\frac{\epsilon^2}{2\lambda}} F(z),$$

and is written by $z \approx_2 \text{prox}_{\lambda F}(y)$. This condition is written equivalently as

$$z \approx_2 \text{prox}_{\lambda F}(y) \Leftrightarrow z \in \left(I + \lambda \partial_{\frac{\epsilon^2}{2\lambda}} F\right)^{-1}.$$

We summarize two accelerated versions of inexact proximal point method using these approximations of proximity operator in Algorithm 11 and 12.

Algorithm 11 Accelerated inexact proximal point method version I

1: **Input** : $y_0 = x_0, t_1 = 1$ and nondecreasing and positive sequence $\{\lambda_k\}$
2: **repeat**
3: Calculate $t_{k+1} = \frac{1+\sqrt{1+4\lambda_k t_k^2/\lambda_{k+1}}}{2}$,
4: Find an approximate minimizer

$$x_{k+1} \approx_1 \text{Prox}_{\lambda_k F}(y_k)$$

5: $y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k)$
6: **until** *a stopping criterion is satisfied.*

When a sequence λ_k satisfies

$$\lambda_j \leq M\lambda_i \quad \text{whenever} \quad j \leq i \text{ for some } M > 0,$$

the convergence of these algorithms was proved in [47] as follows:

Algorithm 12 Accelerated inexact proximal point method version II

1: **Input** : $y_0 = x_0$, arbitrary sequence a_k with $0 < a \leq a_k \leq 2$ and nondecreasing and positive sequence $\{\lambda_k\}$

2: **repeat**

3: Calculate $t_{k+1} = \frac{1+\sqrt{1+4(a_k\lambda_k)t_k^2/(a_{k+1}\lambda_{k+1})}}{2}$,

4: Find an approximate minimizer

$$x_{k+1} \approx_2 \mathrm{Prox}_{\lambda_k F}(y_k)$$

5: $y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k) + (1 - a_k)\frac{t_k}{t_{k+1}}(y_k - x_{k+1})$

6: **until** *a stopping criterion is satisfied.*

- If $\epsilon_k = \mathcal{O}(1/k^q)$ with $q > \frac{3}{2}$, the sequence x_k generated by Algorithm 11 is minimizing for F and if in addition F has a minimizer that following convergence rate holds:

$$F(x_k) - F_* = \begin{cases} \mathcal{O}(1/k^{2q-3}), & \text{if} \quad q < 2 \\ \mathcal{O}(\frac{\log^2 k}{k}), & \text{if} \quad q = 2 \\ \mathcal{O}(1/k), & \text{if} \quad q > 2 \end{cases}.$$

- If $\epsilon_k = \mathcal{O}(1/k^q)$ with $q > \frac{1}{2}$, the sequence x_k generated by Algorithm 12 is minimizing for F and if in addition F has a minimizer that following convergence rate holds:

$$F(x_k) - F_* = \begin{cases} \mathcal{O}(\frac{1}{k^2}), & \text{if} \quad q < \frac{3}{2} \\ \mathcal{O}(\frac{1}{k^2}) + \mathcal{O}(\frac{\log k}{k^2}), & \text{if} \quad q = \frac{3}{2} \\ \mathcal{O}(\frac{1}{k^2}) + \mathcal{O}(\frac{1}{k^{2q-1}}), & \text{if} \quad q > \frac{3}{2} \end{cases}.$$

Chapter 3

Proposed Algorithms

In this chapter, we propose some algorithms which are accelerated version of Bregman method, inexact version of accelerated augmented Lagrangian method and inexact version of accelerated alternating direction method of multipliers. At first, we propose the accelerated Bregman method based on equivalence with the accelerated augmented Lagrangian method which introduced in previous chapter. We also develop the inexact accelerated augmented Lagrangian method(I-AALM) and inexact accelerated alternating direction method of multipliers(I-AADMM). We give the inexact stopping conditions, which can be calculated by numerically, for I-AALM and I-AADMM. For convergence proof for inexact algorithms, we use the technique of convergence proof for inexact version of FISTA in [28] and main convergence theorems of inexact accelerated augmented Lagrangian method and inexact accelerated alternating direction method of multiplier are almost same with that of inexact FISTA. We also represent the numerical tests for proposed algorithms, in last section.

3.1 Proposed Algorithm 1 : Accelerated Bregman method

In this section, we propose an accelerated Bregman method which has an $\mathcal{O}(\frac{1}{k^2})$ global convergence rate. In section 2.2.2, we show that the Bregman method is equivalent to the augmented Lagrangian method. It was proposed in [22] that if $f(x)$ is a differentiable convex function, the augmented Lagrangian method can be accelerated using the extrapolation scheme used in Nesterov's accelerated method. Based on the equivalence and the results in [22], we propose the accelerated Bregman method(Algorithm 13).

Algorithm 13 Accelerated Bregman Method

1: **Input** : $\delta > 0, p_0 = \mathbf{0}$ and $t_0 = 1$
2: **repeat**
3: $\quad x_{k+1} = \arg\min_x D_f^{p_k}(x, x_k) + \frac{\gamma}{2}\|Ax - b\|_2^2$
4: $\quad \tilde{p}_{k+1} = p_k - \gamma A^T(Ax_{k+1} - b)$
5: $\quad t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
6: $\quad p_{k+1} = \tilde{p}_{k+1} + \left(\frac{t_k-1}{t_{k+1}}\right)(\tilde{p}_{k+1} - \tilde{p}_k) + \left(\frac{t_k}{t_{k+1}}\right)(\tilde{p}_{k+1} - p_k)$
7: **until** *a stopping criterion is satisfied.*

3.1.1 Equivalence to the accelerated augmented Lagrangian method

In this subsection, we prove the equivalence between the proposed accelerated Bregman method and the accelerated augmented Lagrangian method which is a generalization of the method proposed in [22] in the sense that f can be nondifferentible. The following lemma is similar to Theorem 2.2.2.

Lemma 3.1.1. *The accelerated Bregman method is equivalent to the following method starting with $b^0 = b$.*

$$\begin{cases} x_{k+1} = \arg\min_x f(x) + \frac{\gamma}{2}\|Ax - b_k\|_2^2 \\ \tilde{b}_{k+1} = b_k - (Ax_{k+1} - b) \\ b_{k+1} = \tilde{b}_{k+1} + \left(\frac{t_k-1}{t_{k+1}}\right)(\tilde{b}_{k+1} - \tilde{b}_k) + \left(\frac{t_k}{t_{k+1}}\right)(\tilde{b}_{k+1} - b_k), \end{cases} \tag{3.1}$$

Proof. We show that (2.5) is satisfied for all k by induction. Based on the initial condition, (2.5) holds for $k = 0$. We assume that (2.5) holds for all

$k \leq n$. For all $k \leq n$, we get

$$\begin{aligned}
\tilde{p}_{k+1} &= p_k - \gamma A^T(Ax_{k+1} - b) \\
&= \gamma A^T(b_k - b) - \gamma A^T(Ax_{k+1} - b) \\
&= \gamma A^T(b_k - b - (Ax_{k+1} - b)) \\
&= \gamma A^T(\tilde{b}_{k+1} - b),
\end{aligned}$$

where the first equality uses the second step of the accelerated Bregman method, the second equality is derived from the induction hypothesis, and the fourth equality uses the second step of (3.1). Hence

$$\tilde{p}_{n+1} = \gamma A^T(\tilde{b}_{n+1} - b).$$

By using the above equality and the third step of (3.1), we have the following equalities

$$\begin{aligned}
p_{n+1} &= \tilde{p}_{n+1} + \left(\frac{t_n - 1}{t_{n+1}}\right)(\tilde{p}_{n+1} - \tilde{p}_n) + \left(\frac{t_n}{t_{n+1}}\right)(\tilde{p}_{n+1} - p_n) \\
&= \gamma A^T(\tilde{b}_{n+1} - b) + \left(\frac{t_n - 1}{t_{n+1}}\right)(\gamma A^T(\tilde{b}_{n+1} - b) - \gamma A^T(\tilde{b}_n - b) \\
&\quad + \left(\frac{t_n}{t_{n+1}}\right)(\gamma A^T(\tilde{b}_{n+1} - b) - \gamma A^T(b_n - b)) \\
&= \gamma A^T\left(\tilde{b}_{n+1} - b + \left(\frac{t_n - 1}{t_{n+1}}\right)(\tilde{b}_{n+1} - \tilde{b}_n) + \left(\frac{t_n}{t_{n+1}}\right)(\tilde{b}_{n+1} - b_n)\right) \\
&= \gamma A^T(b_{n+1} - b).
\end{aligned}$$

By induction,

$$p_k = \gamma A^T(b_{k+1} - b)$$

is satisfied for all integers $k \geq 0$. Thus, the accelerated Bregman method is equivalent to the method (3.1). □

The next lemma shows that the method (3.1) starting with $b_0 = b$ is equivalent to the accelerated augmented Lagrangian method starting with $\lambda_0 = \mathbf{0}$.

Lemma 3.1.2. *The method (3.1) starting with $b_0 = b$ is equivalent to the following accelerated augmented Lagrangian method starting with $\lambda_0 = \boldsymbol{0}$:*

$$\begin{cases} x_{k+1} = \arg\min_x f(x) - (\lambda_k)^T(Ax-b) + \frac{\gamma}{2}\|Ax-b\|_2^2 \\ \tilde{\lambda}_{k+1} = \lambda_k - \gamma(Ax_{k+1} - b) \\ \lambda_{k+1} = \tilde{\lambda}_{k+1} + \left(\frac{t_k-1}{t_{k+1}}\right)(\tilde{\lambda}_{k+1} - \tilde{\lambda}_k) + \left(\frac{t_k}{t_{k+1}}\right)(\tilde{\lambda}_{k+1} - \lambda_k), \end{cases} \tag{3.2}$$

Proof. It is sufficient to show that (2.6) is satisfied for all k according to Lemma 2.2.2. We will prove this by induction. When $k = 0$, (2.6) is satisfied by the initial condition $b_0 = b, \lambda_0 = \mathbf{0}$. We assume that

$$b_k = b + \frac{\lambda_k}{\gamma}$$

is satisfied for $k \leq n$. This implies that, for all $k \leq n$, we have

$$\begin{aligned} \tilde{b}_{k+1} &= b_k - (Ax_{k+1} - b) \\ &= b + \frac{\lambda_k}{\gamma} - (Ax_{k+1} - b) \\ &= b + \frac{1}{\gamma}(\lambda_k - \gamma(Ax_{k+1} - b)) \\ &= b + \frac{\tilde{\lambda}_{k+1}}{\gamma}, \end{aligned}$$

where the first equality is from $\tilde{b}_{k+1}$ in (3.1) and the final equality is from $\tilde{\lambda}_{k+1}$ in (3.2). We find the following equalities using the induction hypothesis and the previous equality

$$\begin{aligned} b_{n+1} &= \tilde{b}_{n+1} + \left(\frac{t_n - 1}{t_{n+1}}\right)(\tilde{b}_{n+1} - \tilde{b}_n) + \left(\frac{t_n}{t_{n+1}}\right)(\tilde{b}_{n+1} - b_n) \\ &= b + \frac{\tilde{\lambda}_{n+1}}{\gamma} + \left(\frac{t_n - 1}{t_{n+1}}\right)\left(b + \frac{\tilde{\lambda}_{n+1}}{\gamma} - b + \frac{\tilde{\lambda}_n}{\gamma}\right) \\ &\quad + \left(\frac{t_n}{t_{n+1}}\right)\left(b + \frac{\tilde{\lambda}_{n+1}}{\gamma} - b + \frac{\lambda_n}{\gamma}\right) \\ &= b + \frac{1}{\gamma}\left(\tilde{\lambda}_{n+1} + \left(\frac{t_n - 1}{t_{n+1}}\right)(\tilde{\lambda}_{n+1} - \tilde{\lambda}_n) + \left(\frac{t_n}{t_{n+1}}\right)(\tilde{\lambda}_{n+1} - \lambda_n)\right) \\ &= b + \frac{\lambda_{n+1}}{\gamma}. \end{aligned}$$

By induction,

$$b^k = b + \frac{\lambda_k}{\gamma}$$

is satisfied for all integers $k \geq 0$. $\square$

By Lemmas 3.1.1 and 3.1.2, we get the following theorem for the equivalence between the accelerated Bregman method and the accelerated augmented Lagrangian method.

Theorem 3.1.1. *The accelerated Bregman method starting with $p_0 = \boldsymbol{0}$ is equivalent to the accelerated augmented Lagrangian method (3.2) starting with $\lambda_0 = \boldsymbol{0}$.*

3.1.2 Complexity of the accelerated Bregman method

In the previous subsection, we proved the equivalence of the accelerated augmented Lagrangian method and the accelerated Bregman method. Therefore, the convergence rate of the (accelerated) Bregman method is equal to the convergence rate of the (accelerated) augmented Lagrangian method. In this section, we will show that the convergence rate of the augmented Lagrangian method (ALM) is $\mathcal{O}(\frac{1}{k})$ and the convergence rate of the accelerated augmented Lagrangian method (AALM) is $\mathcal{O}(\frac{1}{k^2})$.

In [22], the authors considered an augmented Lagrangian method for the linearly constrained smooth minimization problem:

$$\min_{x \in X} \; h(x) \quad \text{s.t.} \quad Ax = b,$$

where h is differentiable and the set X is closed convex set. They showed that augmented Lagrangian method was accelerated and the convergence rate of the accelerated version (3.2) was $\mathcal{O}(\frac{1}{k^2})$. In this paper, we extend the accelerated augmented Lagrangian method in [22] to solve the linearly constrained nonsmooth minimization problem (1.1). The key lemma is Lemma 3.1.4 and the proof of the $\mathcal{O}(\frac{1}{k^2})$ convergence rate is similar to that in [22].

The dual problem of (1.1) is

$$\max_{\lambda} \{ \min_{x} L(x, \lambda) = f(x) - \lambda^T (Ax - b) \}. \tag{3.3}$$

Problem (1.1) is a convex optimization whose constraints are only a linear equation, so there is no duality gap according to Slater's condition [**?**]. Let (x^*, λ^*) be the saddle point of the Lagrangian function. Several lemmas are required to prove the $\mathcal{O}(\frac{1}{k^2})$ convergence rate.

The following lemma gives the bound for the difference of the Lagrangian function values at the current iterates and any point that satisfies $A^T\lambda \in \partial f(x)$ in terms of dual variables.

Lemma 3.1.3. *Let (x_{k+1}, λ_{k+1}) be generated by the augmented Lagrangian method. For any (x, λ) that satisfies*

$$f(x_{k+1}) - f(x) \geq \lambda^T A(x_{k+1} - x), \tag{3.4}$$

we get the inequality

$$L(x_{k+1}, \lambda_{k+1}) - L(x, \lambda) \geq \frac{1}{\gamma}\|\lambda_k - \lambda_{k+1}\|_2^2 + \frac{1}{\gamma}(\lambda - \lambda_k)^T(\lambda_k - \lambda_{k+1}).$$

Proof. By using (3.4) and the definition of the Lagrangian function, we have the following inequalities

$$\begin{aligned}
L(x_{k+1}, \lambda_{k+1}) - L(x, \lambda) &= f(x_{k+1}) - f(x) + \lambda^T(Ax - b) - (\lambda_{k+1})^T(Ax_{k+1} - b) \\
&\geq \lambda^T A(x_{k+1} - x) + \lambda^T(Ax - b) - (\lambda_{k+1})^T(Ax_{k+1} - b) \\
&= \lambda^T(Ax_{k+1} - b) - (\lambda_{k+1})^T(Ax_{k+1} - b) \\
&= (\lambda - \lambda_{k+1})^T(Ax_{k+1} - b) \\
&= \frac{1}{\gamma}(\lambda - \lambda_{k+1})^T(\lambda_k - \lambda_{k+1}) \\
&= \frac{1}{\gamma}\|\lambda_k - \lambda_{k+1}\|_2^2 + \frac{1}{\gamma}(\lambda - \lambda_k)^T(\lambda_k - \lambda_{k+1}),
\end{aligned}$$

where the fourth equality is based on the updating step of λ_{k+1} in the augmented Lagrangian method. □

The next lemma shows that the condition (3.4) in Lemma 3.1.3 is satisfied with $(x, \lambda) = (x_{k+1}, \lambda_{k+1})$, or (x^*, λ^*).

Lemma 3.1.4. *The (x_{n+1}, λ_{n+1}) generated by the augmented Lagrangian method satisfies the condition (3.4) :*

$$f(x_{k+1}) - f(x_{n+1}) \geq (\lambda_{n+1})^T A(x_{k+1} - x_{n+1}),$$

and (x^*, λ^*) *also satisfies the same condition*

$$f(x_{k+1}) - f(x^*) \geq (\lambda^*)^T A(x_{k+1} - x^*).$$

Proof. By Fermat's rule [44, Theorem 10.1] and the third step in the augmented Lagrangian method, we have

$$\mathbf{0} \in \partial f(x_{n+1}) - A^T \lambda_n + \gamma A^T (A x_{n+1} - b),$$

i.e.,

$$A^T(\lambda^n - \gamma(A x_{n+1} - b)) \in \partial f(x_{n+1}).$$

Based on updating rule of λ_{n+1} in the augmented Lagrangian method,

$$A^T \lambda_{n+1} \in \partial f(x_{n+1}).$$

According to the definition of the subdifferential, we get

$$\begin{aligned} f(x_{k+1}) - f(x_{n+1}) &\geq (A^T \lambda_{n+1})^T (x_{k+1} - x_{n+1}) \\ &= (\lambda_{n+1})^T A(x_{k+1} - x_{n+1}). \end{aligned}$$

Since (x^*, λ^*) satisfies the KKT condition,

$$\begin{aligned} \partial f(x^*) - A^T \lambda^* &\ni \mathbf{0} \\ A x^* &= b. \end{aligned}$$

From the first condition, we get

$$A^T \lambda^* \in \partial f(x^*).$$

Thus, based on the definition of the subdifferential, we obtain

$$f(x_{k+1}) - f(x_*) \geq (\lambda^*)^T A(x_{k+1} - x^*).$$

□

By the proof of Lemma 3.1.4, it is satisfied that

$$A^T \lambda_{k+1} \in \partial f(x_{k+1}), \quad A^T \lambda^* \in \partial f(x^*) \quad \text{and} \quad A x^* = b.$$

By the definition of the subdifferential, we get

$$\begin{aligned} f(x^*) - f(x_{k+1}) &\geq (\lambda_{k+1})^T A(x^* - x_{k+1}) \\ &= (\lambda_{k+1})^T (b - Ax_{k+1}). \end{aligned}$$

Hence, we also have

$$L(x_{k+1}, \lambda_{k+1}) \leq L(x^*, \lambda^*).$$

The following lemma is a key lemma to establish the $\mathcal{O}(\frac{1}{k})$ convergence rate for ALM.

Lemma 3.1.5. *Let (x_{k+1}, λ_{k+1}) be generated by the ALM. We have*

$$\|\lambda_{k+1} - \lambda^*\|_2^2 \leq \|\lambda_k - \lambda^*\|_2^2 - \|\lambda_k - \lambda_{k+1}\|_2^2 - 2\gamma(L(x^*, \lambda^*) - L(x_{k+1}, \lambda_{k+1})).$$

Proof. Lemma 3.1.3 with $(x, \lambda) = (x^*, \lambda^*)$ implies that

$$(\lambda_k - \lambda^*)^T (\lambda_k - \lambda_{k+1}) \geq \|\lambda_k - \lambda_{k+1}\|_2^2 + \gamma(L(x^*, \lambda^*) - L(x_{k+1}, \lambda_{k+1})).$$

The above inequality yields that

$$\begin{aligned} \|\lambda_{k+1} - \lambda^*\|_2^2 &= \|\lambda_{k+1} - \lambda_k + \lambda_k - \lambda^*\|_2^2 \\ &= \|\lambda_{k+1} - \lambda_k\|_2^2 - 2(\lambda_k - \lambda^*)^T (\lambda_k - \lambda_{k+1}) + \|\lambda_k - \lambda^*\|_2^2 \\ &\leq \|\lambda_{k+1} - \lambda_k\|_2^2 + \|\lambda_k - \lambda^*\|_2^2 - 2\|\lambda_k - \lambda_{k+1}\|_2^2 \\ &\quad -2\gamma(L(x^*, \lambda^*) - L(x_{k+1}, \lambda_{k+1})) \\ &= \|\lambda_k - \lambda^*\|_2^2 - \|\lambda_k - \lambda_{k+1}\|_2^2 - 2\gamma(L(x^*, \lambda^*) - L(x_{k+1}, \lambda_{k+1})). \end{aligned}$$

□

We have the inequality

$$\|\lambda_{k+1} - \lambda^*\|_2^2 \leq \|\lambda_k - \lambda^*\|_2^2 - \|\lambda_k - \lambda_{k+1}\|_2^2 \tag{3.5}$$

from Lemma 3.1.5 and $L(x_{k+1}, \lambda_{k+1}) \leq L(x^*, \lambda^*)$. Then the inequality (3.5) implies the global convergence of ALM. By summing (3.5) over $k = 1, \cdots, n$ we have

$$\sum_{k=1}^{n} \|\lambda_k - \lambda_{k+1}\|_2^2 \leq \|\lambda_1 - \lambda^*\|_2^2,$$

which implies that

$$\lim_{k \to \infty} \|\lambda_k - \lambda_{k+1}\|_2^2 = 0.$$

In the next theorem, we prove that the convergence rate of ALM is $\mathcal{O}(\frac{1}{k})$.

Theorem 3.1.2. *Let (x_k, λ_k) be generated by the ALM. We obtain*

$$L(x^*, \lambda^*) - L(x_k, \lambda_k) \leq \frac{\|\lambda_0 - \lambda^*\|_2^2}{2k\gamma}.$$

Proof. We get

$$\|\lambda_{n+1} - \lambda^*\|_2^2 \leq \|\lambda_n - \lambda^*\|_2^2 - \|\lambda_n - \lambda_{n+1}\|_2^2 - 2\gamma(L(x^*, \lambda^*) - L(x^{n+1}, \lambda^{n+1}))$$

from Lemma 3.1.5. Thus, we have

$$L(x_{n+1}, \lambda_{n+1}) - L(x^*, \lambda^*) \geq \frac{1}{2\gamma}\left\{\|\lambda_{n+1} - \lambda^*\|_2^2 - \|\lambda_n - \lambda^*\|_2^2 + \|\lambda_n - \lambda_{n+1}\|_2^2\right\}.$$

Summing this inequality over $n = 0, \cdots, k-1$, we have

$$\sum_{n=1}^{k} L(x_n, \lambda_n) - kL(x^*, \lambda^*) \geq \frac{1}{2\gamma}\left\{\|\lambda_k - \lambda^*\|_2^2 - \|\lambda_0 - \lambda^*\|_2^2 + \sum_{n=0}^{k-1}\|\lambda_n - \lambda_{n+1}\|_2^2\right\}. \tag{3.6}$$

Based on Lemma 3.1.3 for $k = n$ and setting $(x, \lambda) = (x_n, \lambda_n)$, we obtain

$$L(x_{n+1}, \lambda_{n+1}) - L(x_n, \lambda_n) \geq \frac{1}{\gamma}\|\lambda_n - \lambda_{n+1}\|_2^2.$$

By multiplying this inequality with n and summing it over $n = 0, \cdots, k-1$, we have the following inequalities

$$\begin{aligned}
& \sum_{n=0}^{k-1} n(L(x_{n+1}, \lambda_{n+1}) - L(x_n, \lambda_n)) \geq \frac{1}{\gamma}\sum_{n=0}^{k-1} n\|\lambda_n - \lambda_{n+1}\|_2^2 \\
\Leftrightarrow\quad & \sum_{n=0}^{k-1}((n+1)L(x_{n+1}, \lambda_{n+1}) - nL(x_n, \lambda_n) - L(x_{n+1}, \lambda_{n+1})) \\
& \qquad\qquad\qquad\qquad \geq \frac{1}{\gamma}\sum_{n=0}^{k-1} n\|\lambda_n - \lambda_{n+1}\|_2^2 \\
\Leftrightarrow\quad & kL(x_k, \lambda_k) - \sum_{n=1}^{k} L(x_n, \lambda_n) \geq \frac{1}{\gamma}\sum_{n=0}^{k-1} n\|\lambda_n - \lambda_{n+1}\|_2^2.
\end{aligned}$$

It follows from adding (3.6) and the above inequality that

$$\begin{aligned}
& k(L(x_k, \lambda_k) - L(x^*, \lambda^*)) \\
& \geq \frac{1}{2\gamma}\left\{\|\lambda_k - \lambda^*\|_2^2 - \|\lambda_0 - \lambda^*\|_2^2 + \sum_{n=0}^{k-1}(2n+1)\|\lambda_n - \lambda_{n+1}\|_2^2\right\}.
\end{aligned}$$

Thus, we have

$$\begin{aligned} L(x^*,\lambda^*) - L(x_k,\lambda_k) &\le \frac{1}{2k\gamma}\Big\{\|\lambda_0-\lambda^*\|_2^2 - \|\lambda_k-\lambda^*\|_2^2 \\ &\qquad - \sum_{n=0}^{k-1}(2n+1)\|\lambda_n-\lambda_{n+1}\|_2^2\Big\} \\ &\le \frac{1}{2k\gamma}\|\lambda_0-\lambda^*\|_2^2. \end{aligned}$$

□

The iterate $(x_{n+1},\tilde{\lambda}_{n+1})$ generated by AALM (3.2) contents

$$f(x_{k+1}) - f(x_{n+1}) \ge (\tilde{\lambda}_{n+1})^T A(x_{k+1}-x_{n+1})$$

by replacing (x_{k+1},λ_{k+1}) with $(x_{k+1},\tilde{\lambda}_{k+1})$ in Lemma 3.1.4. Therefore, we get the following lemmas by simply changing the notation.

Lemma 3.1.6. *Let $(x_{k+1},\tilde{\lambda}_{k+1})$ be generated by the AALM (3.2). For $(x,\lambda) = (x^*,\lambda^*)$ or $(x_{n+1},\tilde{\lambda}_{n+1})$ generated by the AALM, we have the inequality*

$$L(x_{k+1},\tilde{\lambda}_{k+1}) - L(x,\lambda) \ge \frac{1}{\gamma}\|\lambda_k-\tilde{\lambda}_{k+1}\|_2^2 + \frac{1}{\gamma}(\lambda-\lambda_k)^T(\lambda_k-\tilde{\lambda}_{k+1}).$$

Lemma 3.1.7. *Let $(x_{k+1},\tilde{\lambda}_{k+1})$ be generated by the AALM (3.2). We obtain*

$$\|\tilde{\lambda}_{k+1}-\lambda^*\|_2^2 \le \|\lambda_k-\lambda^*\|_2^2 - \|\lambda_k-\tilde{\lambda}_{k+1}\|_2^2 - 2\gamma(L(x^*,\lambda^*) - L(x_{k+1},\tilde{\lambda}_{k+1})).$$

Several lemmas are required to obtain our main result.

Lemma 3.1.8. *The sequence $\{t_k\}$ generated by the accelerated Bregman method satisfies*

$$t_k \ge \frac{k+2}{2}$$

and

$$t_k^2 = t_{k+1}^2 - t_{k+1} \qquad \textit{for all } k \ge 0.$$

Proof. The proof is based on Lemma 3.3 in [22] and the definition of t_k. □

Lemma 3.1.9. *The inequality*

$$4t_k^2 v_{k+1} \leq 4t_0^2 v_1 + \frac{1}{\gamma}\|u_1\|_2^2, \tag{3.7}$$

where $v_k = L(x^*, \lambda^*) - L(x_k, \tilde{\lambda}_k)$ *and* $u_k = t_{k-1}(2\tilde{\lambda}_k - \lambda_{k-1} - \tilde{\lambda}_{k-1}) + \tilde{\lambda}_{k-1} - \lambda^*$, *is satisfied for all* $k \geq 0$

Proof. When $k = 0$, this is trivial. By Lemma 3.1.6 with $(x, \lambda) = (x_n, \tilde{\lambda}_n), (x^*, \lambda^*)$ and using the definition of v_n, we have

$$v_n - v_{n+1} \geq \frac{1}{\gamma}\|\lambda_n - \tilde{\lambda}_{n+1}\|_2^2 + \frac{1}{\gamma}(\tilde{\lambda}_n - \lambda_n)^T(\lambda_n - \tilde{\lambda}_{n+1}) \tag{3.8}$$

$$-v_{n+1} \geq \frac{1}{\gamma}\|\lambda_n - \tilde{\lambda}_{n+1}\|_2^2 + \frac{1}{\gamma}(\lambda^* - \lambda_n)^T(\lambda_n - \tilde{\lambda}_{n+1}). \tag{3.9}$$

By multiplying (3.8) by $t_n - 1$ and adding (3.9), we get

$$(t_n - 1)v_n - t_n v_{n+1} \geq \frac{t_n}{\gamma}\|\lambda_n - \tilde{\lambda}_{n+1}\|_2^2 + \frac{1}{\gamma}(\lambda^* + (t_n - 1)\tilde{\lambda}_n - t_n\lambda_n)^T(\lambda_n - \tilde{\lambda}_{n+1}).$$

By multiplying the above inequality by t_n and applying lemma 3.1.8, we have

$$\begin{aligned}
t_{n-1}^2 v_n - t_n^2 v_{n+1} &\geq \frac{1}{\gamma}(\lambda^* - t_n\lambda_n + (t_n - 1)\tilde{\lambda}_n)^T(t_n(\lambda_n - \tilde{\lambda}_{n+1})) \\
&+ \frac{1}{\gamma}\|t_n(\lambda_n - \tilde{\lambda}_{n+1})\|_2^2 \\
&= \frac{1}{\gamma}(\lambda^* + (t_n - 1)\tilde{\lambda}_n - t_n\tilde{\lambda}_{n+1})^T(t_n(\lambda_n - \tilde{\lambda}_{n+1})) \\
&= \frac{1}{4\gamma}\|\lambda^* + t_n(\tilde{\lambda}_n + \lambda_n - 2\tilde{\lambda}_{n+1}) - \tilde{\lambda}_n\|_2^2 \\
&- \frac{1}{4\gamma}\|\lambda^* + (t_n - 1)\tilde{\lambda}_n - t_n\lambda_n\|_2^2 \\
&= \frac{1}{4\gamma}\|u_{n+1}\|_2^2 - \frac{1}{4\gamma}\|\lambda^* + (t_n - 1)\tilde{\lambda}_n - t_n\lambda_n\|_2^2
\end{aligned}$$

where the second equality is from the elementary identity $x^T y = \frac{1}{4}\|x + y\|_2^2 - \frac{1}{4}\|x - y\|_2^2$. Based on the definition of λ_k in AALM (3.2), we get

$$-\lambda^* - (t_n - 1)\tilde{\lambda}_n + t_n\lambda_n = u_n.$$

Thus, it follows that

$$t_{n-1}^2 v_n - t_n^2 v_{n+1} \geq \frac{1}{4\gamma}\|u_{n+1}\|_2^2 - \frac{1}{4\gamma}\|u_n\|_2^2 \quad \text{for all } n \geq 1. \tag{3.10}$$

By multiplying (3.10) by 4 and summing it over $n = 1, \cdots, k$, we have

$$4(-t_k^2 v_{k+1} + t_0^2 v_1) \geq \frac{1}{\gamma}\|u_{k+1}\|_2^2 - \frac{1}{\gamma}\|u_1\|_2^2.$$

Since $\|u_{k+1}\|_2^2 \geq 0$, we get

$$4t_k^2 v_{k+1} \leq 4t_0^2 v_1 + \frac{1}{\gamma}\|u_1\|_2^2 \quad \text{for all } k \geq 1.$$

□

Now, we establish our main theorem.

Theorem 3.1.3. *Let $(x_{k+1}, \tilde{\lambda}_{k+1}, \lambda_{k+1})$ be generated by the AALM. For any $k \geq 1$, we have*

$$L(x^*, \lambda^*) - L(x_k, \tilde{\lambda}_k) \leq \frac{\|\lambda_0 - \lambda^*\|_2^2}{\gamma(k+1)^2}.$$

Proof. Based on the equation (3.7), we get

$$L(x^*, \lambda^*) - L(x_k, \tilde{\lambda}_k) \leq \frac{4t_0^2 v_1 + \frac{1}{\gamma}\|u_1\|_2^2}{4t_{k-1}^2} \quad \text{for any } k \geq 1.$$

This together with Lemma 3.1.8 implies that

$$L(x^*, \lambda^*) - L(x_k, \tilde{\lambda}_k) \leq \frac{4t_0^2 v_1 + \frac{1}{\gamma}\|u_1\|_2^2}{(k+1)^2}. \tag{3.11}$$

By simple calculation and $t_0 = 1$, we have

$$4t_0^2 v_1 + \frac{1}{\gamma}\|u_1\|_2^2 = 4(L(x^*, \lambda^*) - L(x^1, \tilde{\lambda}^1)) + \frac{1}{\gamma}\|2\tilde{\lambda}_1 - \lambda_0 - \lambda^*\|_2^2.$$

Lemma 3.1.7 with $k = 0$ yields that

$$\|\tilde{\lambda}_1 - \lambda^*\|_2^2 \leq \|\lambda_0 - \lambda^*\|_2^2 - \|\lambda_0 - \tilde{\lambda}_1\|_2^2 - 2\gamma(L(x^*, \lambda^*) - L(x^1, \tilde{\lambda}_1)).$$

Thus, we have

$$
\begin{aligned}
& 4t_0^2 v_1 + \frac{1}{\gamma}\|u_1\|_2^2 \\
\leq \;& \frac{2}{\gamma}\|\lambda_0 - \lambda^*\|_2^2 - \frac{2}{\gamma}\|\lambda_0 - \tilde{\lambda}_1\|_2^2 - \frac{2}{\gamma}\|\tilde{\lambda}_1 - \lambda^*\|_2^2 + \frac{1}{\gamma}\|2\tilde{\lambda}_1 - \lambda_0 - \lambda^*\|_2^2 \\
= \;& \frac{1}{\gamma}\|\lambda_0 - \lambda^*\|_2^2. \qquad (3.12)
\end{aligned}
$$

where the equality is from the identity $2\|a-c\|_2^2 - 2\|b-c\|_2^2 - 2\|b-a\|_2^2 = \|a-c\|_2^2 - \|b-a+b-c\|_2^2$ with $a = \lambda_0, b = \tilde{\lambda}_1, c = \lambda^*$. Thus, (3.11) and (3.12) imply

$$
L(x^*, \lambda^*) - L(x_k, \tilde{\lambda}_k) \leq \frac{\|\lambda_0 - \lambda^*\|_2^2}{\gamma(k+1)^2}.
$$

□

Remark 3.1.1. *In [22], the authors considered a generalized augmented Lagrangian method (3.13) with a symmetric positive definite matrix penalty parameter H_k that satisfied*

$$
H_k \preceq H_{k+1}, \quad \forall k \geq 0
$$

when the object function f was differentiable:

$$
\begin{cases}
x_{k+1} = \arg\min\limits_x \quad f(x) - (\lambda_k)^T(Ax - b) + \frac{1}{2}\|Ax - b\|_{H_k}^2, \\
\lambda_{k+1} = \lambda_k - H_k(Ax_{k+1} - b).
\end{cases} \qquad (3.13)
$$

We can also extend the $\mathcal{O}(\frac{1}{k^2})$ convergence rate result for this generalized method when $f(x)$ is not necessarily differentiable.

3.2 Proposed Algorithm 2 : I-AALM

In this section, we propose an inexact version of the AALM (I-AALM), and we provide an inexact stopping condition of the subproblem with respect to x. The convergence rate is $\mathcal{O}(\frac{1}{k^2})$ like the AALM, although x_k solves inexactly. For comprehension, we write the AALM in Algorithm 14 again by using different notation.

Algorithm 14 AALM

1: **Input** : $\tau > 0, t_0 = 1$ and $\hat{\lambda}_1 = \lambda_0$.

2: **repeat**

3: $\quad x_k = \arg\min_x f(x) - (\hat{\lambda}_k)^T(Ax - b) + \frac{\tau}{2}\|Ax - b\|_2^2,$

4: $\quad \lambda_k = \hat{\lambda}_k - \tau(Ax_k - b),$

5: $\quad t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2},$

6: $\quad \hat{\lambda}_{k+1} = \lambda_k + \frac{t_k - 1}{t_{k+1}}(\lambda_k - \lambda_{k-1}) + \frac{t_k}{t_{k+1}}(\lambda_k - \hat{\lambda}_k)$

7: **until** *a stopping criterion is satisfied.*

We consider the problem (1.1) under the assumption that f is a strongly convex function with the parameter $\sigma_f > 0$. Therefore, the I-AALM to solve (1.1) is given in Algorithm 15.

Algorithm 15 I-AALM

1: **Input** : $\tau > 0, t_0 = 1$ and $\hat{\lambda}_1 = \lambda_0$.

2: **repeat**

3: Find an approximate minimizer

$$x_k \approx \arg\min_x f(x) - (\hat{\lambda}_k)^T(Ax - b) + \frac{\tau}{2}\|Ax - b\|_2^2,$$

satisfying the stopping conditions (3.14).

4: $\quad \lambda_k = \hat{\lambda}_k - \tau(Ax_k - b),$

5: $\quad t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2},$

6: $\quad \hat{\lambda}_{k+1} = \lambda_k + \frac{t_k - 1}{t_{k+1}}(\lambda_k - \lambda_{k-1})$

7: **until** *a stopping criterion is satisfied.*

In this algorithm, the updating step of $\hat{\lambda}$ dose not have an additional term like the updating step of $\hat{\lambda}$ in the AALM (Algorithm 14). In fact, although the updating rule of $\hat{\lambda}_{k+1}$ in the AALM is replaced by $\hat{\lambda}_{k+1} = \lambda_k + \frac{t_k - 1}{t_{k+1}}(\lambda_k - \lambda_{k-1})$, the convergence rate of the algorithm remains $\mathcal{O}\left(\frac{1}{k^2}\right)$ with the number of iterations k.

By Fermat's rule, the optimality condition of the updating step of x_k in the AALM is $0 \in \partial f(x_{k+1}) - A^T\hat{\lambda}_k + \tau A^T(Ax_{k+1} - b)$, i.e.,

$$A^T\lambda_{k+1} = A^T\hat{\lambda}_k - \tau A^T(Ax_{k+1} - b) \in \partial f(x_{k+1}).$$

Based on this note, we introduce the inexact stopping condition for x_k as follows:

$$f'(x_k) - A^T\lambda_k = \delta_k, \|\delta_k\|_2 < \frac{\sigma_f}{\sqrt{\rho(A^TA)}t_k}\epsilon_k \text{ and } \sum_{k=0}^{\infty}\epsilon_k < \infty \text{ with } \epsilon_{k+1} \le \epsilon_k, \tag{3.14}$$

where $f'(x_k)$ is a subgradient of $f(x_k)$. This stopping criterion can be easily computed, so it can be directly used in numerical experiments. On the other hand, the related work [23, 32] considering an inexact solution of the subproblem cannot compute a stopping criterion, since their stopping criterions involves a true solution. In the numerical section, we provide a stopping criterion for each application.

By the stopping condition (3.14) and the property (2.2), we can derive the following relations

$$A^T\lambda_k + \delta_k \in \partial f(x_k) \Rightarrow x_k \in \nabla f^*(A^T\lambda_k + \delta_k) \Rightarrow Ax_k \in A\nabla f^*(A^T\lambda_k + \delta_k). \tag{3.15}$$

The last condition is the main property useful to prove the convergence of our I-AALM. Additionally, we assume that $\max_{\lambda} D(\lambda)$ is achieved at λ^*. In the following, several lemmas are presented to prove that the convergence rate is $\mathcal{O}(\frac{1}{k^2})$.

The following lemma gives the bound for the difference of the Lagrangian dual function values at the current iterates and any point in terms of dual variables:

Lemma 3.2.1. *Let $(\lambda_{k+1}, \hat{\lambda}_{k+1})$ be generated by I-AALM. For any $\gamma \in \mathbb{R}^m$ and $k \ge 0$, we have*

$$\begin{aligned} D(\lambda_{k+1}) - D(\gamma) \quad \ge \quad & \frac{1}{\tau}(\gamma - \hat{\lambda}_{k+1})^T(\hat{\lambda}_{k+1} - \lambda_{k+1}) + \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 \\ & +(\gamma - \lambda_{k+1})^T\eta_{k+1}, \end{aligned}$$

where $\eta_{k+1} = A\nabla f^*(A^T\lambda_{k+1}) - A\nabla f^*(A^T\lambda_{k+1} + \delta_{k+1})$.

Proof. We easily verify the following inequality.

$$f^*(A^T\gamma) - f^*(A^T\lambda_{k+1}) \geq (\gamma - \lambda_{k+1})^T(A\nabla f^*(A^T\lambda_{k+1})). \tag{3.16}$$

By using (3.16) and (3.15), we have

$$\begin{aligned}
D(\lambda_{k+1}) - D(\gamma) &= f^*(A^T\gamma) - f^*(A^T\lambda_{k+1}) + (\lambda_{k+1} - \gamma)^T b \\
&\geq (\gamma - \lambda_{k+1})^T(A\nabla f^*(A^T\lambda_{k+1})) + (\lambda_{k+1} - \gamma)^T b \\
&= (\gamma - \lambda_{k+1})^T(Ax_{k+1} - b) + (\gamma - \lambda_{k+1})^T\eta_{k+1} \\
&= \frac{1}{\tau}(\gamma - \lambda_{k+1})^T(\hat{\lambda}_{k+1} - \lambda_{k+1}) + (\gamma - \lambda_{k+1})^T\eta_{k+1} \\
&= \frac{1}{\tau}(\gamma - \hat{\lambda}_{k+1})^T(\hat{\lambda}_{k+1} - \lambda_{k+1}) + \frac{1}{\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 \\
&\quad + (\gamma - \lambda_{k+1})^T\eta_{k+1} \\
&\geq \frac{1}{\tau}(\gamma - \hat{\lambda}_{k+1})^T(\hat{\lambda}_{k+1} - \lambda_{k+1}) + \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 \\
&\quad + (\gamma - \lambda_{k+1})^T\eta_{k+1},
\end{aligned}$$

where the second inequality is from (3.16), the third equality is from (3.15) and the fourth equality is from updating step of λ_k in I-AALM. □

We obtain the following lemma by simple calculation and the updating step of $\hat{\lambda}_{k+1}$ in I-AALM.

Lemma 3.2.2. *It is satisfied that* $s_{k+1} = s_k + t_{k+1}(\lambda_{k+1} - \hat{\lambda}_{k+1})$, *where* $s_k = t_k\lambda_k - (t_k - 1)\lambda_{k-1} - \lambda^*$.

Proof. Using the update rule of I-AALM

$$\hat{\lambda}_{k+1} = \lambda_k + \frac{t_k - 1}{t_{k+1}}(\lambda_k - \lambda_{k-1}),$$

we get following equalities:

$$\begin{aligned}
s_{k+1} &= t_{k+1}\lambda_{k+1} - (t_{k+1} - 1)\lambda_k - \lambda^* \\
&= \lambda_k - \lambda^* + t_{k+1}(\lambda_{k+1} - \lambda_k) \\
&= \lambda_k - (t_k - 1)\lambda_{k-1} - \lambda^* + t_{k+1}(\lambda_{k+1} - \lambda_k) + (t_k - 1)\lambda_{k-1} \\
&= t_k\lambda_k - (t_k - 1)\lambda_{k-1} - \lambda^* + t_{k+1}(\lambda_{k+1} - \lambda_k) + (t_k - 1)(\lambda_{k-1} - \lambda_k) \\
&= s_k + t_{k+1}(\lambda_{k+1} - \lambda_k) - t_{k+1}(\hat{\lambda}_{k+1} - \lambda_k) \\
&= s_k + t_{k+1}(\lambda_{k+1} - \hat{\lambda}_{k+1}).
\end{aligned}$$

□

Several lemmas are required to obtain our main result.

Lemma 3.2.3. *Under same notation in Lemma 3.2.2, we have*

$$\begin{aligned}\|s_{k+1}\|^2 - \|s_k\|^2 &\leq 2t_k^2\tau(D(\lambda^*) - D(\lambda_k)) - 2t_{k+1}^2\tau(D(\lambda^*) - D(\lambda_{k+1}))\\ &\quad +2\tau t_{k+1}(s_{k+1})^T\eta_{k+1}.\end{aligned}$$

Proof. By Lemma 3.2.2, we get

$$\begin{aligned}\|s_{k+1}\|^2 - \|s_k\|^2 &= 2t_{k+1}s_k^T(\lambda_{k+1} - \hat{\lambda}_{k+1}) + t_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2\\ &= 2t_{k+1}(t_k\lambda_k - (t_k - 1)\lambda_{k-1} - \lambda^*)^T(\lambda_{k+1} - \hat{\lambda}_{k+1})\\ &\quad +t_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2.\end{aligned}$$

From updating rule of $\hat{\lambda}_{k+1}$ in I-AALM, note that

$$(t_k - 1)(\lambda_k - \lambda_{k-1}) + \lambda_k = t_{k+1}\hat{\lambda}_{k+1} - t_{k+1}\lambda_k + \lambda_k = t_{k+1}\hat{\lambda}_{k+1} + (1 - t_{k+1})\lambda_k.$$

This note yields

$$\begin{aligned}\|s_{k+1}\|^2 - \|s_k\|^2 &= 2t_{k+1}(t_{k+1}\hat{\lambda}_{k+1} + (1 - t_{k+1})\lambda_k - \lambda^*)^T(\lambda_{k+1} - \hat{\lambda}_{k+1})\\ &\quad +t_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2\\ &= 2t_{k+1}((1 - t_{k+1})(\lambda_k - \hat{\lambda}_{k+1}) + \hat{\lambda}_{k+1} - \lambda^*)^T(\lambda_{k+1} - \hat{\lambda}_{k+1})\\ &\quad +t_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2\\ &= 2t_{k+1}(1 - t_{k+1})(\lambda_k - \hat{\lambda}_{k+1})^T(\lambda_{k+1} - \hat{\lambda}_{k+1})\\ &\quad +2t_{k+1}(\hat{\lambda}_{k+1} - \lambda^*)^T(\lambda_{k+1} - \hat{\lambda}_{k+1}) + t_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2\\ &= 2(t_{k+1}^2 - t_{k+1})\Big\{(\hat{\lambda}_{k+1} - \lambda_k)^T(\lambda_{k+1} - \hat{\lambda}_{k+1})\\ &\quad +\frac{1}{2}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2\Big\} + 2t_{k+1}\Big\{(\hat{\lambda}_{k+1} - \lambda^*)^T(\lambda_{k+1} - \hat{\lambda}_{k+1})\\ &\quad +\frac{1}{2}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2\Big\}.\end{aligned}$$

By Lemma 3.2.1 with setting $\gamma = \lambda_k$ and λ^*, we obtain

$$\begin{aligned}D(\lambda_{k+1}) - D(\lambda_k) &\geq \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 + \frac{1}{\tau}(\lambda_k - \hat{\lambda}_{k+1})^T(\hat{\lambda}_{k+1} - \lambda_{k+1})\\ &\quad +(\lambda_k - \lambda_{k+1})^T\eta_{k+1}\end{aligned}$$

and

$$\begin{aligned} D(\lambda_{k+1}) - D(\lambda^*) &\geq \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 + \frac{1}{\tau}(\lambda^* - \hat{\lambda}_{k+1})^T(\hat{\lambda}_{k+1} - \lambda_{k+1}) \\ &\quad +(\lambda^* - \lambda_{k+1})^T\eta_{k+1}. \end{aligned}$$

From above inequalities and $t_{k+1}^2 - t_{k+1} = t_k^2$ in Lemma 3.1.8, we have

$$\begin{aligned} \|s_{k+1}\|_2^2 - \|s_k\|_2^2 &\leq 2\tau t_{k+1}(t_{k+1}-1)(D(\lambda_{k+1}) - D(\lambda_k)) \\ &\quad +2\tau t_{k+1}(D(\lambda_{k+1}) - D(\lambda^*)) \\ &\quad -2\tau t_{k+1}(t_{k+1}-1)(\lambda_k - \lambda_{k+1})^T\eta_{k+1} \\ &\quad -2\tau t_{k+1}(\lambda^* - \lambda_{k+1})^T\eta_{k+1} \\ &= 2\tau t_{k+1}^2 D(\lambda_{k+1}) - 2\tau t_k^2 D(\lambda_k) - 2\tau(t_{k+1}^2 - t_k^2)D(\lambda^*) \\ &\quad -2\tau t_k^2(\lambda_k - \lambda_{k+1})^T\eta_{k+1} - 2\tau t_{k+1}(\lambda^* - \lambda_{k+1})^T\eta_{k+1} \\ &= 2\tau t_{k+1}^2(D(\lambda_{k+1}) - D(\lambda^*)) - 2\tau t_k^2(D(\lambda_k) - D(\lambda^*)) \\ &\quad -2\tau t_k^2(\lambda_k - \lambda_{k+1})^T\eta_{k+1} - 2\tau t_{k+1}(\lambda^* - \lambda_{k+1})^T\eta_{k+1}. \end{aligned}$$

We get the following by simple calculation and Lemma 3.1.8:

$$\begin{aligned} 2t_k^2(\lambda_k - \lambda_{k+1}) + 2t_{k+1}(\lambda^* - \lambda_{k+1}) &= 2t_k^2\lambda_k - 2t_k^2\lambda_{k+1} + 2t_{k+1}\lambda^* - 2t_{k+1}\lambda_{k+1} \\ &= 2t_{k+1}\lambda^* - 2t_{k+1}^2\lambda_{k+1} + 2t_k^2\lambda_k \\ &= 2t_{k+1}(\lambda^* - t_{k+1}\lambda_{k+1} + (t_{k+1}-1)\lambda_k) \\ &= -2t_{k+1}s_{k+1}. \end{aligned}$$

Finally, we have the last equation

$$\begin{aligned} \|s_{k+1}\|_2^2 - \|s_k\|_2^2 &\leq 2\tau t_{k+1}^2(D(\lambda_{k+1}) - D(\lambda^*)) - 2\tau t_k^2(D(\lambda_k) - D(\lambda^*)) \\ &\quad +2\tau t_{k+1}(s_{k+1})^T\eta_{k+1}. \end{aligned}$$

□

Now we present our main theorem including the convergence rate of our proposed algorithm I-AALM. The proof of this theorem is motivated by inexact accelerated proximal gradient method [28].

Theorem 3.2.1. *Let $(\lambda_{k+1}, \hat{\lambda}_{k+1})$ be generated by I-AALM. We have the following inequality :*

$$t_k^2(D(\lambda^*) - D(\lambda_k)) \leq \left(\sqrt{2\tau}\bar{\epsilon}_k + \frac{1}{\sqrt{2\tau}}\|\lambda^* - \hat{\lambda}_1\|_2\right)^2 + 2\tilde{\epsilon}_k,$$

where $\tilde{\epsilon}_k = 2\tau \sum_{j=1}^{k} \epsilon_j^2$ and $\bar{\epsilon}_k = \sum_{j=1}^{k} \epsilon_j$. That is,

$$D(\lambda^*) - D(\lambda_k) \leq \frac{4}{(k+1)^2}\left[\left(\sqrt{2\tau}\bar{\epsilon}_k + \frac{1}{\sqrt{2\tau}}\|\lambda^* - \hat{\lambda}_1\|_2\right)^2 + 2\tilde{\epsilon}_k\right].$$

Proof. Let $h_k = t_k^2(D(\lambda^*) - D(\lambda_k)) \geq 0$ and $p_k = \frac{1}{2\tau}\|s_k\|_2^2$. By Lemma 3.2.1 with setting $\gamma = \lambda^*$ and $k = 0$, we have

$$\begin{aligned} -h_1 &\geq \frac{1}{\tau}(\lambda^* - \hat{\lambda}_1)^T(\hat{\lambda}_1 - \lambda_1) + \frac{1}{2\tau}\|\hat{\lambda}_1 - \lambda_1\|_2^2 + (\lambda^* - \lambda_1)^T\eta_1 \\ &= \frac{1}{2\tau}\|\lambda_1 - \lambda^*\|_2^2 - \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 + (\lambda^* - \lambda_1)^T\eta_1 \\ &= p_1 - \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 - (s_1)^T\eta_1. \end{aligned} \tag{3.17}$$

Note that

$$(s_k)^T\eta_k \leq \|s_k\|_2\|\eta_k\|_2 \leq \|s_k\|_2\left(\frac{\sqrt{\rho(A^TA)}}{\sigma_f}\|\delta_k\|_2\right) \leq \|s_k\|_2\frac{\epsilon_k}{t_k} = \sqrt{2\tau p_k}\frac{\epsilon_k}{t_k},$$

from the inexact stopping condition (3.14). Hence, from the inequality (3.17), we have

$$h_1 + p_1 \leq \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 + \epsilon_1\sqrt{2\tau p_1}. \tag{3.18}$$

By Lemma 3.2.3 and the inequality (3.18) that considers the upper bound of $(s_k)^T\eta_k$, we have

$$h_{k+1} + p_{k+1} \leq h_k + p_k + \sqrt{2\tau p_{k+1}}\epsilon_{k+1}. \tag{3.19}$$

Using (3.18) and (3.19), we obtain

$$\begin{aligned} \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 &\geq h_1 + p_1 - \epsilon_1\sqrt{2\tau p_1} \\ &\geq h_2 + p_2 - \epsilon_1\sqrt{2\tau p_1} - \epsilon_2\sqrt{2\tau p_2} \\ &\geq \cdots \geq h_k + p_k - q_k, \end{aligned} \tag{3.20}$$

where $q_k = \sqrt{2\tau p_1}\epsilon_1 + \cdots + \sqrt{2\tau p_k}\epsilon_k$.

Since $h_k \geq 0$,then we get the following by the inequality (3.20)

$$q_k = q_{k-1} + \epsilon_k\sqrt{2\tau p_k} \leq q_{k-1} + \epsilon_k\sqrt{2\tau\left(q_k + \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2\right)}. \tag{3.21}$$

Since $\frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 \geq p_1 - \epsilon_1\sqrt{2\tau p_1}$, we have the following inequalities, using the triangle inequality

$$\sqrt{p_1} \leq \frac{\epsilon_1\sqrt{2\tau} + \sqrt{2\tau\epsilon_1^2 + 4(\frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2)}}{2} \leq \epsilon_1\sqrt{2\tau} + \frac{1}{\sqrt{2\tau}}\|\lambda^* - \hat{\lambda}_1\|_2.$$

This yields

$$q_1 = \sqrt{2\tau p_1}\epsilon_1 \leq \sqrt{2\tau}\epsilon_1\left[\epsilon_1\sqrt{2\tau} + \frac{1}{\sqrt{2\tau}}\|\lambda^* - \hat{\lambda}_1\|_2\right] = 2\tau\epsilon_1^2 + \epsilon_1\|\lambda^* - \hat{\lambda}_1\|_2. \tag{3.22}$$

And from (3.21), we have

$$\begin{aligned}\left(\frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 + q_k\right) - \epsilon_k\sqrt{2\tau\left(q_k + \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2\right)}& \\ - \left(\frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 + q_{k-1}\right) &\leq 0,\end{aligned}$$

i.e.

$$\sqrt{q_k + \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2} \leq \frac{1}{2}\left[\sqrt{2\tau}\epsilon_k + \sqrt{2\tau\epsilon_k^2 + 4\left(\frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 + q_{k-1}\right)}\right]. \tag{3.23}$$

Consequently, we obtain

$$\begin{aligned} q_k &\leq q_{k-1} + \epsilon_k^2\tau + \frac{1}{2}\epsilon_k\sqrt{2\tau}\sqrt{2\tau\epsilon_k^2 + 4\left(\frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 + q_{k-1}\right)} \\ &\leq q_{k-1} + 2\epsilon_k^2\tau + \epsilon_k\left(\|\lambda^* - \hat{\lambda}_1\|_2 + \sqrt{2\tau q_{k-1}}\right), \end{aligned}$$

where the first inequality is obtained from (3.21) and (3.23), and the second inequality is due to the triangle inequality.

By summing the above inequality from 2 to k, we have

$$\begin{aligned}
q_k &\leq q_1 + 2\tau \sum_{j=2}^{k} \epsilon_j^2 + \|\lambda^* - \hat{\lambda}_1\|_2 \sum_{j=2}^{k} \epsilon_j + \sum_{j=2}^{k} \epsilon_j \sqrt{2\tau q_{j-1}} \\
&\leq 2\tau \sum_{j=1}^{k} \epsilon_j^2 + \|\lambda^* - \hat{\lambda}_1\|_2 \sum_{j=1}^{k} \epsilon_j + \sum_{j=1}^{k} \epsilon_j \sqrt{2\tau q_j} \\
&\leq 2\tau \sum_{j=1}^{k} \epsilon_j^2 + \|\lambda^* - \hat{\lambda}_1\|_2 \sum_{j=1}^{k} \epsilon_j + \sqrt{2\tau q_k} \sum_{j=1}^{k} \epsilon_j \\
&= \|\lambda^* - \hat{\lambda}_1\|_2 \bar{\epsilon}_k + \tilde{\epsilon}_k + \sqrt{2\tau q_k}\bar{\epsilon}_k
\end{aligned}$$

where the second inequality is according to (3.22), and $\epsilon_j \geq \epsilon_{j+1}$, for all j. This implies that

$$\sqrt{q_k} \leq \frac{1}{2}\left(\sqrt{2\tau}\bar{\epsilon}_k + \sqrt{2\tau\bar{\epsilon}_k^2 + 4\|\lambda^* - \hat{\lambda}_1\|_2\bar{\epsilon}_k + 4\tilde{\epsilon}_k}\right).$$

Therefore, we have $q_k \leq 2\tau\bar{\epsilon}_k^2 + 2\|\lambda^* - \hat{\lambda}_1\|_2\bar{\epsilon}_k + 2\tilde{\epsilon}_k$, by the arithmetic mean-geometric mean inequality.

Since $h_k \leq \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 + q_k$, we derive the final conclusion as

$$h_k \leq \left(\sqrt{2\tau}\bar{\epsilon}_k + \frac{1}{\sqrt{2\tau}}\|\lambda^* - \hat{\lambda}_1\|_2\right)^2 + 2\tilde{\epsilon}_k.$$

□

3.3 Proposed Algorithm 3 : I-AADMM

In this section, we propose an inexact version of the accelerated ADMM (FADMM) [18], so called I-AADMM, with inexact stopping conditions for the subproblems. Moreover, we prove that the convergence rate of our algorithm remains $\mathcal{O}(\frac{1}{k^2})$ for each iteration k. We consider the problem (1.2) assuming that H is a strongly convex function with σ_H and G is a quadratic strongly convex function with σ_G.

Algorithm 16 I-AADMM

Input : $\tau > 0, t_0 = 1, \lambda_0 = \hat{\lambda}_1, v_0 = \hat{v}_1$ satisfying the equation (3.26) and a sequence ϵ_k satisfying (3.29)

repeat

Find an approximate minimizer

$$u_k \approx \arg\min_u H(u) - (\hat{\lambda}_k)^T(Bu) + \frac{\tau}{2}\|Bu + C\hat{v}_k - b\|_2^2, \tag{3.24}$$

satisfying the stopping conditions (3.30).

Find an approximate minimizer

$$v_k \approx \arg\min_v G(v) - (\hat{\lambda}_k)^T(Cv) + \frac{\tau}{2}\|Bu_k + Cv - b\|_2^2, \tag{3.25}$$

satisfying the stopping conditions (3.31).

$\lambda_k = \hat{\lambda}_k - \tau(Bu_k + Cv_k - b)$,

$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$,

$\hat{v}_{k+1} = v_k + \frac{t_k - 1}{t_{k+1}}(v_k - v_{k-1})$,

$\hat{\lambda}_{k+1} = \lambda_k + \frac{t_k - 1}{t_{k+1}}(\lambda_k - \lambda_{k-1})$

until *a stopping criterion is satisfied.*

In this algorithm, we take initial variables satisfying the following condition:

$$C^T\hat{\lambda}_1 + \xi_0 \in \partial G(\hat{v}_1) \text{ with } \|\xi_0\|_2 \leq \frac{\sqrt{2\tau\rho(C^TC)}}{\sqrt{6}t_2}\epsilon_0. \tag{3.26}$$

Let $\tilde{\lambda}_k = \hat{\lambda}_k - \tau(Bu_k + C\hat{v}_k - b)$. Then, by Fermat's rule, the optimality conditions of the subproblems with respect to u and v in the I-AADMM are as follows

$$0 \in \partial H(u_k) - B^T\hat{\lambda}_k + \tau B^T(Bu_k + C\hat{v}_{k-1} - b) = \partial H(u_k) - B^T\tilde{\lambda}_k \tag{3.27}$$

$$0 \in \partial G(v_k) - C^T\hat{\lambda}_k + \tau C^T(Bu_k + Cv_k - b) = \partial G(v_k) - C^T\lambda_k. \tag{3.28}$$

First, let us consider a sequence ϵ_k satisfying

$$\sum_{k=0}^{\infty}\epsilon_k < \infty, \quad \text{with} \quad \epsilon_{k+1} \leq \epsilon_k. \tag{3.29}$$

For the subproblem (3.24) of u_k, we introduce the following stopping conditions:

$$h'(u_k) - B^T\tilde{\lambda}_k = \delta_k \text{ with } \|\delta_k\|_2 < \frac{\sigma_H}{2\sqrt{\rho(B^TB)}t_k}\epsilon_k \tag{3.30}$$

where $h'(u_k)$ is a subgradient of $H(u^k)$. For the subproblem (3.25) of v_k, the proposed stopping condition is given by

$$g'(v_k) - C^T\lambda_k = \xi_k \text{ with } \|\xi_k\|_2 < \min\left\{\frac{\sqrt{2\tau\rho(C^TC)}}{\sqrt{6}t_{k+2}}, \frac{\sigma_G}{2\sqrt{\rho(C^TC)}t_k}\right\}\epsilon_k \tag{3.31}$$

where $g'(v_k)$ is a subgradient of $G(v^k)$.

The stopping conditions (3.30), (3.31) with the properties of subdifferential derive the following relations

$$B^T\tilde{\lambda}_k + \delta_k \in \partial H(u_k) \Rightarrow u_k \in \nabla H^*(B^T\tilde{\lambda}_k + \delta_k) \Rightarrow Bu_k \in B\nabla H^*(B^T\tilde{\lambda}_k + \delta_k), \tag{3.32}$$

$$C^T\lambda_k + \delta_k \in \partial G(v_k) \Rightarrow v_k \in \nabla G^*(C^T\lambda_k + \xi_k) \Rightarrow Cv_k \in C\nabla G^*(C^T\lambda_k + \xi_k). \tag{3.33}$$

Since G is strongly convex and quadratic, ∇G^* is an affine transformation. Hence,

$$\begin{aligned} C\hat{v}_{k+1} &= Cv_k + \frac{t_k - 1}{t_{k+1}}(Cv_k - Cv_{k-1}) \\ &\in C\nabla G^* \left(C^T \hat{\lambda}_{k+1} + \frac{t_{k+1} + t_k - 1}{t_{k+1}} \xi_k - \frac{t_k - 1}{t_{k+1}} \xi_{k-1} \right) \end{aligned} \quad (3.34)$$

Based on the notes (3.27)-(3.34), we prove the convergence rate of the I-AADMM as the similar way with the I-AALM. We also assume that maximization problem of Lagrangian dual function of the problem (1.2) is achieved at λ^*.

Lemma 3.3.1. *Suppose that* $\tau^3 \leq \dfrac{\sigma_H \sigma_G^2}{\rho(B^T B)\rho(C^T C)^2}$. *Then for any* $\gamma \in \mathbb{R}^m$ *and* $k \geq 1$, *we have*

$$\begin{aligned} D(\lambda_{k+1}) - D(\gamma) &\geq \frac{1}{\tau}(\gamma - \hat{\lambda}_{k+1})^T(\hat{\lambda}_{k+1} - \lambda_{k+1}) + \frac{1}{2\tau}\|\hat{\lambda}_{k+1} - \lambda_{k+1}\|_2^2 \\ &- \frac{1}{t_{k+1}^2}\epsilon_{k-1}^2 + (\gamma - \lambda_{k+1})^T(\eta_{k+1}^1 + \eta_{k+1}^2), \end{aligned}$$

where $(\lambda_{k+1}, \hat{\lambda}_{k+1})$ *are generated by I-AADMM,*

$$\eta_{k+1}^1 = C\nabla G^*(C^T\lambda_{k+1}) - C\nabla G^*(C^T\lambda_{k+1} + \xi_{k+1})$$

and

$$\eta_{k+1}^2 = B\nabla H^*(B^T\tilde{\lambda}_{k+1}) - B\nabla H^*(B^T\tilde{\lambda}_{k+1} + \delta_{k+1}).$$

Proof. Set $\alpha = \dfrac{\rho(B^T B)}{\sigma_H}$ and $\beta = \dfrac{\rho(C^T C)}{\sigma_G}$. We get the following inequalities:

$$\begin{aligned} \|\tilde{\lambda}_{k+1} - \lambda_{k+1}\|_2^2 &= \tau^2\|C\hat{v}_{k+1} - Cv_{k+1}\|_2^2 \\ &= \tau^2 \left\| C\nabla G^* \left(C^T\hat{\lambda}_{k+1} + \frac{t_{k+1} + t_k - 1}{t_{k+1}}\xi_k - \frac{t_k - 1}{t_{k+1}}\xi_{k-1} \right) \right. \\ &\quad \left. - C\nabla G^*(C^T\lambda_{k+1} + \xi_{k+1}) \right\|_2^2 \\ &\leq \frac{\tau^2\rho(C^T C)}{\sigma_G^2} \left\| C^T\hat{\lambda}_{k+1} + \frac{t_{k+1} + t_k - 1}{t_{k+1}}\xi_k - \frac{t_k - 1}{t_{k+1}}\xi_{k-1} \right. \\ &\quad \left. - C^T\lambda_{k+1} - \xi_{k+1} \right\|_2^2 \end{aligned}$$

$$
\begin{aligned}
&\leq \tau^2\beta^2\|\lambda_{k+1}-\hat{\lambda}_{k+1}\|_2^2+\frac{\tau^2\rho(C^TC)}{\sigma_G^2}\left(\frac{(t_k-1)^2}{t_{k+1}^2}\|\xi_{k-1}\|_2^2\right.\\
&\quad\left.+\frac{(t_{k+1}+t_k-1)^2}{t_{k+1}^2}\|\xi_k\|_2^2+\|\xi_{k+1}\|_2^2\right),\\
&\leq \tau^2\beta^2\|\lambda_{k+1}-\hat{\lambda}_{k+1}\|_2^2+\frac{2\tau^3\rho(C^TC)^2}{\sigma_G^2t_{k+1}^2}\epsilon_{k-1}^2 \qquad (3.35)
\end{aligned}
$$

where the first equality is from definition of $\tilde{\lambda}_{k+1}$ and updating rule of λ_{k+1} in I-AADMM, the second equality is from (3.33) and (3.34), the third inequality is from Lipschitz continuous of ∇G^* with Lipschitz constant $1/\sigma_G^2$, the fourth inequality is from triangle inequality and the last inequality is from following notes:

- $t_{i+1}\geq t_i$ and $\epsilon_{i+1}\leq\epsilon_i$ for all $i=1,2,...$
- $\dfrac{(t_{k+1}+t_k-1)^2}{t_{k+1}^2}\leq 4$ and $\frac{(t_k-1)^2}{t_{k+1}^2}\leq 1$.
- $\max\left\{\|\xi_{k-1}\|_2^2,\|\xi_k\|_2^2,\|\xi_{k+1}\|_2^2\right\}\leq\dfrac{2\tau\rho(C^TC)}{6t_{k+1}^2}\epsilon_{k-1}^2$.

By strongly convexity of H, we obtain

$$
\begin{aligned}
H^*(B^T\gamma)-H^*(B^T\lambda_{k+1}) &= H^*(B^T\gamma)-H^*(B^T\tilde{\lambda}_{k+1})\\
&\quad +H^*(B^T\tilde{\lambda}_{k+1})-H^*(B^T\lambda_{k+1})\\
&\geq (\gamma-\tilde{\lambda}_{k+1})^T(B\nabla H^*(B^T\tilde{\lambda}_{k+1}))-\frac{\alpha}{2}\|\lambda_{k+1}-\tilde{\lambda}_{k+1}\|_2^2\\
&\quad -(\lambda_{k+1}-\tilde{\lambda}_{k+1})^T(B\nabla H^*(B^T\tilde{\lambda}_{k+1}))\\
&= (\gamma-\lambda_{k+1})^T(B\nabla H^*(B^T\tilde{\lambda}_{k+1}))\\
&\quad -\frac{\alpha\tau^2\beta^2}{2}\|\lambda_{k+1}-\hat{\lambda}_{k+1}\|_2^2-\frac{\alpha\tau^3\rho(C^TC)^2}{\sigma_G^2t_{k+1}^2}\epsilon_{k-1}^2\\
&\geq (\gamma-\lambda_{k+1})^T(B\nabla H^*(B^T\tilde{\lambda}_{k+1}))\\
&\quad -\frac{1}{2\tau}\|\lambda_{k+1}-\hat{\lambda}_{k+1}\|_2^2-\frac{1}{t_{k+1}^2}\epsilon_{k-1}^2, \qquad (3.36)
\end{aligned}
$$

where the third equality is from (3.35) and the last inequality is from the assumption $\tau^3 \leq \dfrac{\sigma_H \sigma_G^2}{\rho(A^T A)\rho(B^T B)^2}$. Similarly, we get

$$G^*(C^T\gamma) - G^*(C^T\lambda_{k+1}) \geq (\gamma - \lambda_{k+1})^T (C\nabla G^*(C^T\lambda_{k+1})). \tag{3.37}$$

Using (3.36) and (3.37), we have

$$\begin{aligned}
D(\lambda_{k+1}) - D(\gamma) &= G^*(C^T\gamma) - G^*(C^T\lambda_{k+1}) + H^*(B^T\gamma) - H^*(B^T\lambda_{k+1}) \\
&\quad +(\lambda_{k+1} - \gamma)^T b \\
&\geq (\gamma - \lambda_{k+1})^T (B\nabla H^*(B^T\tilde{\lambda}_{k+1})) - \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 \\
&\quad +(\gamma - \lambda_{k+1})^T (C\nabla G^*(C^T\lambda_{k+1})) + (\lambda_{k+1} - \gamma)^T b - \frac{1}{t_{k+1}^2}\epsilon_{k-1}^2 \\
&= (\gamma - \lambda_{k+1})^T (Bu_{k+1} + Cv_{k+1} - b) - \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 \\
&\quad -\frac{1}{t_{k+1}^2}\epsilon_{k-1}^2 + (\gamma - \lambda_{k+1})^T (\eta_{k+1}^1 + \eta_{k+1}^2) \\
&= \frac{1}{\tau}(\gamma - \lambda_{k+1})^T (\hat{\lambda}_{k+1} - \lambda_{k+1}) - \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 \\
&\quad -\frac{1}{t_{k+1}^2}\epsilon_{k-1}^2 + (\gamma - \lambda_{k+1})^T (\eta_{k+1}^1 + \eta_{k+1}^2) \\
&= \frac{1}{\tau}(\gamma - \hat{\lambda}_{k+1})^T (\hat{\lambda}_{k+1} - \lambda_{k+1}) + \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 \\
&\quad -\frac{1}{t_{k+1}^2}\epsilon_{k-1}^2 + (\gamma - \lambda_{k+1})^T (\eta_{k+1}^1 + \eta_{k+1}^2),
\end{aligned}$$

□

Remark 3.3.1. *If* $k = 0$, $C\hat{v}_0 \in C\partial G^*(C^T\hat{\lambda}_1 + \xi_0)$. *Hence, we have by similar way to proof of Lemma 3.3.1*

$$\begin{aligned}
D(\lambda_1) - D(\gamma) &\geq \tau^{-1}(\gamma - \hat{\lambda}_1)^T (\hat{\lambda}_1 - \lambda_1) + \frac{1}{2\tau}\|\hat{\lambda}_1 - \lambda_1\|_2^2 \\
&- \frac{1}{t_1^2}\epsilon_0^2 + (\gamma - \lambda_1)^T (\eta_1^1 + \eta_1^2).
\end{aligned}$$

In proof, we use simple property :

$$\|\xi_0\|_2 \leq \frac{\sqrt{2\tau\rho(C^T C)}}{\sqrt{6}t_2}\epsilon_0 \leq \frac{\sqrt{2\tau\rho(C^T C)}}{2t_1}\epsilon_0$$

Since the updating rule of $\hat{\lambda}_{k+1}$ in I-AADMM is same that in I-AALM, the following Lemma is obvious.

Lemma 3.3.2. *It is satisfied that*

$$s_{k+1} = s_k + t_{k+1}(\lambda_{k+1} - \hat{\lambda}_{k+1}),$$

where $s_k = t_k\lambda_k - (t_k - 1)\lambda_{k-1} - \lambda^*$ *and* $(\lambda_k, \hat{\lambda}_k)$ *is generated by I-AADMM.*

The following Lemma is similar with Lemma 3.2.3, but for completion, we provide the full proof of Lemma 3.3.3.

Lemma 3.3.3. *Under same notation in Lemma 3.3.2, we have*

$$\begin{aligned}\|s_{k+1}\|^2 - \|s_k\|^2 \ \leq \ & 2t_k^2\tau(D(\lambda^*) - D(\lambda_k)) - 2t_{k+1}^2\tau(D(\lambda^*) - D(\lambda_{k+1})) \\ & +2\tau\epsilon_{k-1}^2 + 2t_{k+1}(s_{k+1})^T(\eta_{k+1}^1 + \eta_{k+1}^2).\end{aligned}$$

Proof. By Lemma 3.3.2, we have the equations:

$$\begin{aligned}\|s_{k+1}\|^2 - \|s_k\|^2 &= 2t_{k+1}s_k^T(\lambda_{k+1} - \hat{\lambda}_{k+1}) + t_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 \\ &= 2t_{k+1}(t_k\lambda_k - (t_k - 1)\lambda_{k-1} - \lambda^*)^T(\lambda_{k+1} - \hat{\lambda}_{k+1}) \\ &\quad +t_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2\end{aligned}$$

From updating rule of $\hat{\lambda}_{k+1}$ in I-AADMM, note that

$$(t_k - 1)(\lambda_k - \lambda_{k-1}) + \lambda_k = t_{k+1}\hat{\lambda}_{k+1} - t_{k+1}\lambda_k + \lambda_k = t_{k+1}\hat{\lambda}_{k+1} + (1 - t_{k+1})\lambda_k.$$

From this note, we have

$$\begin{aligned}\|s_{k+1}\|^2 - \|s_k\|^2 &= 2t_{k+1}(t_{k+1}\hat{\lambda}_{k+1} + \lambda_k(1 - t_{k+1}) - \lambda^*)^T(\lambda_{k+1} - \hat{\lambda}_{k+1}) \\ &\quad +t_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 \\ &= 2t_{k+1}((1 - t_{k+1})(\lambda_k - \hat{\lambda}_{k+1}) + \hat{\lambda}_{k+1} - \lambda^*)^T(\lambda_{k+1} - \hat{\lambda}_{k+1}) \\ &\quad +t_{k+1}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 \\ &= 2t_{k+1}(1 - t_{k+1})(\lambda_k - \hat{\lambda}_{k+1})^T(\lambda_{k+1} - \hat{\lambda}_{k+1}) \\ &\quad +2t_{k+1}(\hat{\lambda}_{k+1} - \lambda^*)^T(\lambda_{k+1} - \hat{\lambda}_{k+1}) \\ &\quad +t_{k+1}^2\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 \\ &= 2(t_{k+1}^2 - t_{k+1})\Big\{(\hat{\lambda}_{k+1} - \lambda_k)^T(\lambda_{k+1} - \hat{\lambda}_{k+1}) \\ &\quad +\frac{1}{2}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2\Big\} + 2t_{k+1}\Big\{(\hat{\lambda}_{k+1} - \lambda^*)^T(\lambda_{k+1} - \hat{\lambda}_{k+1}) \\ &\quad +\frac{1}{2}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2\Big\}.\end{aligned}$$

By Lemma 3.3.1 with setting $\gamma = \lambda_k$ and λ^*, we obtain

$$\begin{aligned} D(\lambda_{k+1}) - D(\lambda_k) \;\geq\;& \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 + \frac{1}{\tau}(\lambda_k - \hat{\lambda}_{k+1})^T(\hat{\lambda}_{k+1} - \lambda_{k+1}) \\ & - \frac{1}{t_{k+1}^2}\epsilon_{k-1}^2 + (\lambda_k - \lambda_{k+1})^T(\eta_{k+1}^1 + \eta_{k+1}^2) \end{aligned}$$

and

$$\begin{aligned} D(\lambda_{k+1}) - D(\lambda^*) \;\geq\;& \frac{1}{2\tau}\|\lambda_{k+1} - \hat{\lambda}_{k+1}\|_2^2 + \frac{1}{\tau}(\lambda^* - \hat{\lambda}_{k+1})^T(\hat{\lambda}_{k+1} - \lambda_{k+1}) \\ & - \frac{1}{t_{k+1}^2}\epsilon_{k-1}^2 + (\lambda^* - \lambda_{k+1})^T(\eta_{k+1}^1 + \eta_{k+1}^2). \end{aligned}$$

From above inequalities and $t_{k+1}^2 - t_{k+1} = t_k^2$, we have

$$\begin{aligned} \|s_{k+1}\|_2^2 - \|s_k\|_2^2 \;\leq\;& 2\tau t_{k+1}(t_{k+1} - 1)(D(\lambda_{k+1}) - D(\lambda_k)) \\ & + 2\tau t_{k+1}(D(\lambda_{k+1}) - D(\lambda^*)) \\ & + 2\tau t_k^2 \left\{ \frac{\epsilon_{k-1}^2}{t_{k+1}^2} - (\lambda_k - \lambda_{k+1})^T(\eta_{k+1}^1 + \eta_{k+1}^2) \right\} \\ & + 2\tau t_{k+1} \left\{ \frac{\epsilon_{k-1}^2}{t_{k+1}^2} - (\lambda^* - \lambda_{k+1})^T(\eta_{k+1}^1 + \eta_{k+1}^2) \right\} \\ =\;& 2\tau t_{k+1}^2 D(\lambda_{k+1}) - 2\tau t_k^2 D(\lambda_k) - 2(t_{k+1}^2 - t_k^2)D(\lambda^*) \\ & + 2\tau t_k^2 \left\{ \frac{\epsilon_{k-1}^2}{t_{k+1}^2} - (\lambda_k - \lambda_{k+1})^T(\eta_{k+1}^1 + \eta_{k+1}^2) \right\} \\ & + 2\tau t_{k+1} \left\{ \frac{\epsilon_{k-1}^2}{t_{k+1}^2} - (\lambda^* - \lambda_{k+1})^T(\eta_{k+1}^1 + \eta_{k+1}^2) \right\} \\ =\;& 2\tau t_{k+1}^2(D(\lambda_{k+1}) - D(\lambda^*)) - 2t_k^2\tau(D(\lambda_k) - D(\lambda^*)) \\ & + 2\tau t_k^2 \left\{ \frac{\epsilon_{k-1}^2}{t_{k+1}^2} - (\lambda_k - \lambda_{k+1})^T(\eta_{k+1}^1 + \eta_{k+1}^2) \right\} \\ & + 2\tau t_{k+1} \left\{ \frac{\epsilon_{k-1}^2}{t_{k+1}^2} - (\lambda^* - \lambda_{k+1})^T(\eta_{k+1}^1 + \eta_{k+1}^2) \right\}. \end{aligned}$$

We can change the terms of ϵ_{k-1}^2 and $\eta_{k+1}^1 + \eta_{k+1}^2$ in above inequality simply as follows:

$$2t_k^2\left(\frac{1}{t_{k+1}^2}\right)\epsilon_{k-1}^2 + 2t_{k+1}\left(\frac{1}{t_{k+1}^2}\right)\epsilon_{k-1}^2 = 2(t_k^2 + t_{k+1})\left(\frac{1}{t_{k+1}^2}\right)\epsilon_{k-1}^2 = 2\epsilon_{k-1}^2$$

and

$$\begin{aligned}
2t_k^2(\lambda_k - \lambda_{k+1}) + 2t_{k+1}(\lambda^* - \lambda_{k+1}) &= 2t_k^2\lambda_k - 2t_k^2\lambda_{k+1} + 2t_{k+1}\lambda^* - 2t_{k+1}\lambda_{k+1} \\
&= 2t_{k+1}\lambda^* - 2t_{k+1}^2\lambda_{k+1} + 2t_k^2\lambda_k \\
&= 2t_{k+1}(\lambda^* - t_{k+1}\lambda_{k+1} + (t_{k+1} - 1)\lambda_k) \\
&= -2t_{k+1}s_{k+1}.
\end{aligned}$$

Thus, we have the final equation in this proof:

$$\begin{aligned}
\|s_{k+1}\|_2^2 - \|s_k\|_2^2 \leq\ & 2t_{k+1}^2\tau(D(\lambda_{k+1}) - D(\lambda^*)) - 2t_k^2\tau(D(\lambda_k) - D(\lambda^*)) \\
& +2\tau\epsilon_{k-1}^2 + 2t_{k+1}(s_{k+1})^T(\eta_{k+1}^1 + \eta_{k+1}^2).
\end{aligned}$$

□

The following theorem is our main theorem which represents the convergence rate of the I-AADMM.

Theorem 3.3.1. *Let $(\lambda_k, \hat{\lambda}_k)$ be generated by I-AADMM. Then, we have*

$$t_k^2(D(\lambda^*) - D(\lambda_k)) \leq \left(\sqrt{2\tau}\bar{\epsilon}_k + \frac{1}{\sqrt{2\tau}}\|\lambda^* - \hat{\lambda}_1\|_2\right)^2 + 2\tilde{\epsilon}_k,$$

where $\tilde{\epsilon}_k = \sum_{j=1}^{k}\epsilon_{j-2}^2 + 2\tau\sum_{j=1}^{k}\epsilon_j^2 + \sqrt{2\tau}\sum_{j=1}^{k}\epsilon_j\epsilon_{j-2}, \bar{\epsilon}_k = \sum_{j=1}^{k}\epsilon_j$ *and* $\epsilon_{-1} = \epsilon_0$*. It means that*

$$D(\lambda^*) - D(\lambda_k) \leq \frac{4}{(k+1)^2}\left[\left(\sqrt{2\tau}\bar{\epsilon}_k + \frac{1}{\sqrt{2\tau}}\|\lambda^* - \hat{\lambda}_1\|_2\right)^2 + 2\tilde{\epsilon}_k\right].$$

Proof. Let $h_k = t_k^2(D(\lambda^*) - D(\lambda_k)) \geq 0$ and $p_k = \frac{1}{2\tau}\|s_k\|_2^2$. By Remark 3.3.1 with setting $\gamma = \lambda^*$, we have

$$\begin{aligned}
-h_1 &\geq \frac{1}{\tau}(\lambda^* - \hat{\lambda}_1)^T(\hat{\lambda}_1 - \lambda_1) + \frac{1}{2\tau}\|\hat{\lambda}_1 - \lambda_1\|_2^2 - \frac{1}{t_2^2}\epsilon_0^2 + (\lambda^* - \lambda_1)^T(\eta_1^1 + \eta_1^2) \\
&= \frac{1}{2\tau}\|\lambda_1 - \lambda^*\|_2^2 - \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 - \frac{1}{t_2^2}\epsilon_0^2 + (\lambda^* - \lambda_1)^T(\eta_1^1 + \eta_1^2) \\
&= p_1 - \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 - \frac{1}{t_2^2}\epsilon_0^2 + (s_1)^T(\eta_1^1 + \eta_1^2). \qquad (3.38)
\end{aligned}$$

From inexact conditions (3.30) and (3.31) of subproblems in the I-AADMM, we note that

$$\begin{aligned}
(s_k)^T(\eta_k^1+\eta_k^2) &\leq \|s_k\|_2(\|\eta_k^1\|_2+\|\eta_k^2\|_2) \\
&\leq \|s_k\|_2\left(\frac{\sqrt{\rho(B^TB)}}{\sigma_H}\|\delta_k\|_2+\frac{\sqrt{\rho(C^TC)}}{\sigma_G}\|\xi_k\|_2\right) \\
&\leq \|s_k\|_2\frac{\epsilon_k}{t_k} \\
&\leq \sqrt{2\tau p_k}\frac{\epsilon_k}{t_k}.
\end{aligned}$$

Hence, from the inequality (3.38) and setting $\epsilon_{-1}=\epsilon_0$, we obtain

$$h_1+p_1 \leq \frac{1}{2\tau}\|\lambda^*-\hat{\lambda}_1\|_2^2+\epsilon_0^2+\epsilon_1\sqrt{2\tau p_1} = \frac{1}{2\tau}\|\lambda^*-\hat{\lambda}_1\|_2^2+\epsilon_{-1}^2+\epsilon_1\sqrt{2\tau p_1}. \quad (3.39)$$

By Lemma 3.3.3 and the inequality (3.39) about the upper bound of $(s_k)^T(\eta_k^1+\eta_k^2)$, we have

$$h_{k+1}+p_{k+1} \leq h_k+p_k+\epsilon_{k-1}^2+\sqrt{2\tau p_{k+1}}\epsilon_{k+1}. \quad (3.40)$$

The equations (3.39) and (3.40) yield

$$\begin{aligned}
\frac{1}{2\tau}\|\lambda^*-\hat{\lambda}_1\|_2^2 &\geq h_1+p_1-\epsilon_{-1}^2-\epsilon_1\sqrt{2\tau p_1} \\
&\geq h_2+p_2-\epsilon_{-1}^2-\epsilon_0^2-\epsilon_1\sqrt{2\tau p_1}-\epsilon_2\sqrt{2\tau p_2} \\
&\geq \cdots \geq h_k+p_k-q_k, \quad (3.41)
\end{aligned}$$

where $q_k=\sqrt{2\tau p_1}\epsilon_1+\cdots+\sqrt{2\tau p_k}\epsilon_k+\epsilon_{-1}^2+\cdots+\epsilon_{k-2}^2$. Since $h_k \geq 0$, we obtain the following, by the inequality (3.41)

$$\begin{aligned}
q_k &= q_{k-1}+\epsilon_k\sqrt{2\tau p_k}+\epsilon_{k-2}^2 \\
&\leq q_{k-1}+\epsilon_k\sqrt{2\tau\left(q_k+\frac{1}{2\tau}\|\lambda^*-\hat{\lambda}_1\|_2^2\right)}+\epsilon_{k-2}^2. \quad (3.42)
\end{aligned}$$

Since $\frac{1}{2\tau}\|\lambda^*-\hat{\lambda}_1\|_2^2 \geq p_1-\epsilon_{-1}^2-\epsilon_1\sqrt{2\tau p_1}$, we get the following, by the triangle inequality

$$\begin{aligned}
\sqrt{p_1} &\leq \frac{\epsilon_1\sqrt{2\tau}+\sqrt{2\tau\epsilon_1^2+4(\epsilon_{-1}^2+\frac{1}{2\tau}\|\lambda^*-\hat{\lambda}_1\|_2^2)}}{2} \\
&\leq \epsilon_1\sqrt{2\tau}+\sqrt{\epsilon_{-1}^2+\frac{1}{2\tau}\|\lambda^*-\hat{\lambda}_1\|_2^2}.
\end{aligned}$$

This inequality and the triangle inequality yield

$$
\begin{aligned}
q_1 = \sqrt{2\tau p_1}\epsilon_1 + \epsilon_{-1}^2 &\leq \sqrt{2\tau}\epsilon_1 \left[\epsilon_1\sqrt{2\tau} + \sqrt{\epsilon_{-1}^2 + \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2}\right] + \epsilon_{-1}^2 \\
&\leq 2\tau\epsilon_1^2 + \epsilon_{-1}^2 + \epsilon_1(\sqrt{2\tau}\epsilon_{-1} + \|\lambda^* - \hat{\lambda}_1\|_2). \qquad (3.43)
\end{aligned}
$$

From (3.42), we obtain

$$
\begin{aligned}
\left(\frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 + q_k\right) - \epsilon_k\sqrt{2\tau\left(q_k + \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2\right)}& \\
-\left(\frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 + q_{k-1} + \epsilon_{k-2}^2\right) &\leq 0,
\end{aligned}
$$

i.e.,

$$
\sqrt{q_k + \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2} \leq \frac{1}{2}\left[\sqrt{2\tau}\epsilon_k + \sqrt{2\tau\epsilon_k^2 + 4\left(\frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 + q_{k-1} + \epsilon_{k-2}^2\right)}\right]. \qquad (3.44)
$$

Consequently, we obtain the following inequalities

$$
\begin{aligned}
q_k &\leq q_{k-1} + \epsilon_{k-2}^2 + \epsilon_k^2\tau + \frac{1}{2}\epsilon_k\sqrt{2\tau}\sqrt{2\tau\epsilon_k^2 + 4\left(\frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 + q_{k-1} + \epsilon_{k-2}^2\right)} \\
&\leq q_{k-1} + \epsilon_{k-2}^2 + 2\epsilon_k^2\tau + \epsilon_k\left(\|\lambda^* - \hat{\lambda}_1\|_2 + \sqrt{2\tau q_{k-1}} + \sqrt{2\tau}\epsilon_{k-2}\right),
\end{aligned}
$$

where the first inequality is from (3.42) and (3.44), and the last inequality is from the triangle inequality. By summing the inequality (3.45) from 2 to k, we have

$$
\begin{aligned}
q_k &\leq q_1 + \sum_{j=2}^{k}\epsilon_{j-2}^2 + 2\tau\sum_{j=2}^{k}\epsilon_j^2 + \|\lambda^* - \hat{\lambda}_1\|_2\sum_{j=2}^{k}\epsilon_j + \sqrt{2\tau}\sum_{j=2}^{k}\epsilon_j(\sqrt{q_{j-1}} + \epsilon_{j-2}) \\
&\leq \sum_{j=1}^{k}\epsilon_{j-2}^2 + 2\tau\sum_{j=1}^{k}\epsilon_j^2 + \|\lambda^* - \hat{\lambda}_1\|_2\sum_{j=1}^{k}\epsilon_j + \sum_{j=1}^{k}\epsilon_j\sqrt{2\tau q_j} + \sqrt{2\tau}\sum_{j=1}^{k}\epsilon_j\epsilon_{j-2} \\
&\leq \sum_{j=1}^{k}\epsilon_{j-2}^2 + 2\tau\sum_{j=1}^{k}\epsilon_j^2 + \|\lambda^* - \hat{\lambda}_1\|_2\sum_{j=1}^{k}\epsilon_j + \sqrt{2\tau q_k}\sum_{j=1}^{k}\epsilon_j + \sqrt{2\tau}\sum_{j=1}^{k}\epsilon_j\epsilon_{j-2} \\
&\leq \|\lambda^* - \hat{\lambda}_1\|_2\bar{\epsilon}_k + \tilde{\epsilon}_k + \sqrt{2\tau q_k}\bar{\epsilon}_k
\end{aligned}
$$

where the second inequality is from (3.43) and $\epsilon_j \leq \epsilon_{j+1}$, which implies that

$$\sqrt{q_k} \leq \frac{1}{2}\left(\sqrt{2\tau}\bar{\epsilon}_k + \sqrt{2\tau\bar{\epsilon}_k^2 + 4\|\lambda^* - \hat{\lambda}_1\|_2\bar{\epsilon}_k + 4\tilde{\epsilon}_k}\right).$$

From here, we have $q_k \leq 2\tau\bar{\epsilon}_k^2 + 2\|\lambda^* - \hat{\lambda}_1\|_2\bar{\epsilon}_k + 2\tilde{\epsilon}_k$, by the arithmetic mean-geometric mean inequality. Since $h_k \leq \frac{1}{2\tau}\|\lambda^* - \hat{\lambda}_1\|_2^2 + q_k$, we have

$$h_k \leq \left(\sqrt{2\tau}\bar{\epsilon}_k + \frac{1}{\sqrt{2\tau}}\|\lambda^* - \hat{\lambda}_1\|_2\right)^2 + 2\tilde{\epsilon}_k.$$

□

3.4 Numerical Results

In this section, we provide the numerical test applying our proposed algorithms which are the accelerated Bregman method, the I-AALM and the I-AADMM. In subsection 3.4.1, we perform the numerical test making a comparison between Bregman method and accelerated Bregman method for solving the linearly constrained ℓ_1 and generalized ℓ_2 minimization. For applying the I-AALM, we solve the linearly constrained ℓ_1-ℓ_2 minimization problem in subsection 3.4.2 and subsection 3.4.3. In subsection 3.4.2, we use the various algorithms for solving the subproblem of I-AALM and confirm the convergence of I-AALM although the subproblem is inexactly solved. In subsection 3.4.3, we compare the performance of I-AALM with state-of-art algorithms for solving the linearly constrained ℓ_1-ℓ_2 minimization. Lastly, we propose the new variational model for removing multiplicative noise and we apply our I-AADMM to this new model. We compare the denoising results of our new model with TV model.

3.4.1 Comparison to Bregman method with accelerated Bregman method

In this subsection, we compare the performance of the Bregman method with that of the accelerated Bregman method for solving the linearly constrained

ℓ_1 and generalized ℓ_2 minimization :

$$\min_x \mu\|x\|_1 + \frac{1}{2}x^T Qx \quad \text{subject to} \quad Ax = b, \tag{3.45}$$

where $\|\cdot\|_1$ is the ℓ_1-norm in $\mathbb{R}^n$, $\|\cdot\|_2$ is the ℓ_2-norm in $\mathbb{R}^n$, $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^m$ and Q is a symmetric positive definite matrix with size $n \times n$. Note that $x^T Qx > 0$ for all nonzero vector $x \in \mathbb{R}^n$, since Q is a symmetric positive definite matrix. In addition, we easily prove that $x^T Qx$ is a norm when Q is a symmetric positive definite matrix. Thus, we can write $\|x\|_Q = \sqrt{x^T Qx}$ and it is called generalized ℓ_2-norm with respect to Q.

If $Q = \mathbf{0}$, the equation (3.45) is basis pursuit problem in compressive sensing. The object function of the basis pursuit is not strongly convex function. Adding the $\|\cdot\|_Q$ in the basis pursuit problem yields the tractable object function $\mu\|x\|_1 + \|x\|_Q^2$, which is a strongly convex function. Thus, the linearly constrained ℓ_1 and generalized ℓ_2 minimization problem (3.45) has a unique solution and the dual problem is smooth. Problem (3.45) has a generalized ℓ_2-norm in the regularizer term, so it is less sensitive to noise than the basis pursuit problem. Actually, many researchers considered this model (3.45) in [8, **?**, 56] related to linearized Bregman method and compressive sensing when $Q = \beta I$. We consider this case $Q = \beta I$ in next subsection. We set $Q = \lambda I - \gamma A^T A$ in these experiments. If $\lambda > \gamma\|A\|_2^2$, Q is a symmetric positive definite. In this case, the subproblem with respect to x in Bregman method or accelerated Bregman method

$$\arg\min_x \mu\|x\|_1 + \frac{1}{2}\|x\|_Q^2 - p_k^T x + \frac{\gamma}{2}\|Ax - b\|_2^2$$

has closed form solution

$$x = \text{shrink}\left(\frac{p_k + \gamma A^T b}{\lambda}, \frac{\mu}{\lambda}\right),$$

where the soft thresholding or shrinkage operator is defined by

$$(\text{shrink}(x, \alpha))_i = \text{sign}(x_i)(|x_i| - \alpha)_+.$$

For experiments, we fix the parameters $\lambda = 2$, $\gamma = 1$ and $\mu = 10$. we set $n = 1000$, $m = 500$ for the size of the Gaussian measurement matrix A

whose entries are selected randomly from a standard Gaussian distribution $\mathcal{N}(0,1)$ with $\|A^TA\|_2 = 1$. For making $\|A^TA\|_2 = 1$, we use the following matlab codes :

```
opts_eigs.issym = 1; opts_eigs.disp = 0;
eig_sqrt = sqrt(eigs(A*A',1,'lm',opts_eigs));
A = A/eig_sqrt;
```

Since $\|A^TA\|_2 = 1$, $Q = \lambda I - \gamma A^TA$ is a symmetric positive definite matrix. The sparsity k, i.e., the number of nonzero elements of the original solution, is fixed at 50. The location of nonzero elements in the original solution (signal) $\bar{x}$ is selected randomly and the nonzero elements of $\bar{x}$ are selected from uniform distribution in interval $[-10, 10]$. (matlab code : 20*(rand(k,1)-0.5)). The noise n in

$$b = A\bar{x} + n$$

is generated by a standard Gaussian distribution $\mathcal{N}(0,1)$ and then it is normalized to the norm $\sigma = 0, 0.1, 1$.(When $\sigma = 0$, we consider the noise-free case.) When noise is present, we terminate all the methods when the residual error $\|Ax^k - b\|_2 \le \sigma$. But, for a noise-free case, i.e., $b = A\bar{x}$ or $\sigma = 0$, we stop all the methods when the residual error

$$\|Ax^k - b\|_2 < 10^{-4}$$

is satisfied. In each test, we calculate the residual error $\|Ax - b\|_2$, the relative error

$$\frac{\|x - \bar{x}\|_2}{\|\bar{x}\|_2},$$

and the signal-to-noise ratio (SNR)

$$10 \log_{10} \frac{\|\bar{x} - \ \mathrm{mean}(\bar{x})\|_2^2}{\|\bar{x} - x\|_2^2},$$

where x is the recovery signal. For this setting, 100 different tests are conducted.

In Figures 3.1 - 3.3, we plot the Lagrangian function value, relative error and residual error at each iteration of the accelerated Bregman method and Bregman method in 1st row for various σ. We also plot the comparison final

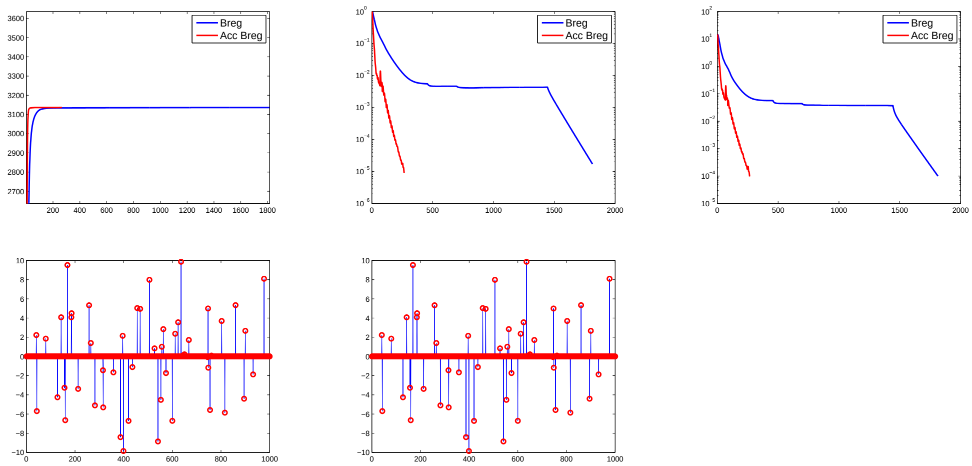

Figure 3.1: Results from linearly constrained ℓ_1 and generalized ℓ_2 minimization problem for noise-free case. 1st row : Lagrangian function (1st column), relative error (2nd column) and residual error (3rd column) at each iteration. 2nd row : Plot the recover vector (blue) for Bregman method and the solution vector (red) in 1st column and Plot the recover vector (blue) for accelerated Bregman method and the solution vector (red) in 2nd column

Table 3.1: Comparison of the Bregman method with the accelerated Bregman method for noise-free case

		mean	std.	maximum	minimum
Iteration	Bregman	2132.71	2621.94	24460	660
	Acc. Bregman	240.94	50.46	532	189
Time	Bregman	16.7696	20.3191	188.3310	5.2542
	Acc. Bregman	1.8917	0.3945	4.0727	1.4680
Res. Err.	Bregman	9.954e-05	3.200e-07	9.999e-05	9.868e-05
	Acc. Bregman	9.315e-05	5.162e-06	9.996e-05	7.953e-05
Rel. Err.	Bregman	1.950e-05	3.826e-06	3.622e-05	1.206e-05
	Acc. Bregman	7.135e-06	6.793e-07	8.763e-06	5.781e-06
SNR	Bregman	94.349	1.612	98.36	88.804
	Acc. Bregman	102.966	0.829	104.76	101.15

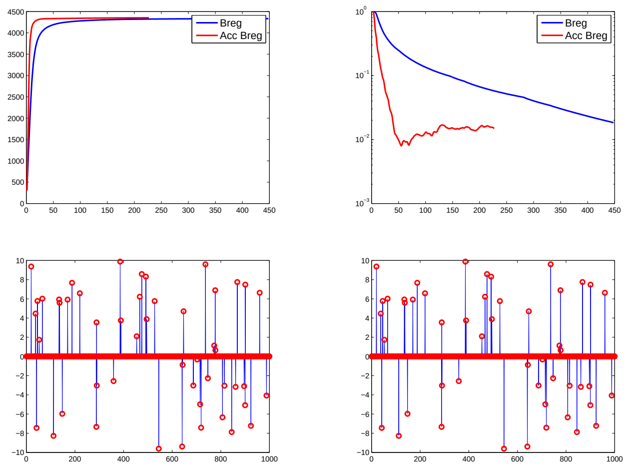

Figure 3.2: Results from linearly constrained ℓ_1 and generalized ℓ_2 minimization problem for $\sigma = 0.1$. 1st row : Lagrangian function (1st column), relative error (2nd column) and residual error (3rd column) at each iteration. 2nd row : Plot the recover vector (blue) for Bregman method and the solution vector (red) in 1st column and Plot the recover vector (blue) for accelerated Bregman method and the solution vector (red) in 2nd column

Table 3.2: Comparison of the Bregman method with the accelerated Bregman method for $\sigma = 0.1$.

		mean	std.	maximum	minimum
Iteration	Bregman	486.34	146.91	933.00	228.00
	Acc Bregman	240.37	28.25	288.00	72.00
Time	Bregman	3.8230	1.1714	7.2579	1.7712
	Acc Bregman	1.8840	0.2269	2.3225	0.5463
Res. Err.	Bregman	9.972e-02	3.444e-04	1.000e-01	9.806e-02
	Acc Bregman	9.825e-02	1.570e-03	9.998e-02	9.322e-02
Rel. Err.	Bregman	1.264e-02	3.437e-03	2.401e-02	7.202e-03
	Acc Bregman	1.594e-02	1.821e-03	2.083e-02	8.275e-03
SNR	Bregman	38.261	2.271	42.851	32.392
	Acc Bregman	36.004	1.046	41.641	33.621

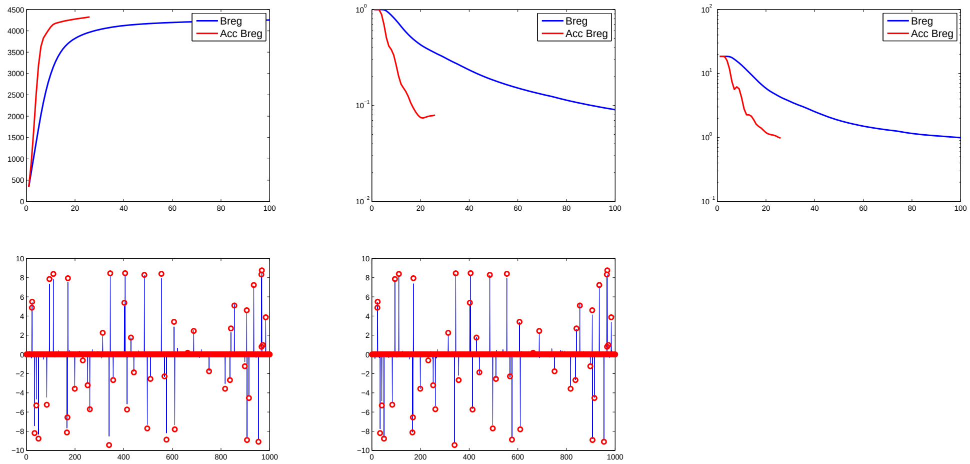

Figure 3.3: Results from linearly constrained ℓ_1 and generalized ℓ_2 minimization problem for $\sigma = 1$. 1st row : Lagrangian function (1st column), relative error (2nd column) and residual error (3rd column) at each iteration. 2nd row : Plot the recover vector (blue) for Bregman method and the solution vector (red) in 1st column and Plot the recover vector (blue) for accelerated Bregman method and the solution vector (red) in 2nd column

Table 3.3: Comparison of the Bregman method with the accelerated Bregman method for $\sigma = 1$.

		mean	std.	maximum	minimum
Iteration	Bregman	113.95	12.23	156.00	91.00
	Acc Bregman	29.00	2.53	38.00	25.00
Time	Bregman	0.9133	0.1013	1.3204	0.7283
	Acc Bregman	0.2251	0.0220	0.2876	0.1863
Res. Err.	Bregman	9.966e-01	2.550e-03	1.000e+00	9.888e-01
	Acc Bregman	9.804e-01	1.711e-02	9.999e-01	9.326e-01
Rel. Err.	Bregman	1.154e-01	1.590e-02	1.639e-01	6.946e-02
	Acc Bregman	8.744e-02	1.212e-02	1.192e-01	6.550e-02
SNR	Bregman	18.834	1.228	23.165	15.705
	Acc Bregman	21.242	1.186	23.676	18.471

recovery vector with original exact solution for each method in 2nd row. In Tables 3.1 - 3.3, we report the average number of iterations, the CPU time, the residual error, the relative error, and SNR of recovery solution for various noise. From number of iterations and CPU time in Tables 3.1 - 3.3, the accelerated Bregman method is faster than the Bregman method for all noise level. Based on the relative error and SNR in Table 3.1, it is observed that the sparse original signal is well restored, especially, the recovery solution of the accelerated Bregman method is more accurate than that of the Bregman method when $\sigma = 0$. This is also shown in Figure 3.1. We conclude that the accelerated Bregman method has a better performance of both speed and accuracy than the Bregman method for noise-free case. Meanwhile, the observed solution of the Bregman method is more accurate than that of the accelerated Bregman method when $\sigma = 0.1$. In Figure 3.2, the residual error of accelerated Bregma method stays around 0.1 and the relative error of accelerated Bregma method increases when number of iterations is between 50 and 200. Since the accelerated Bregman method is not monotone method and noise is present, these problems can be present. If we find more suitable stopping criteria, these problems can be improved. Based on 2nd row of Figure 3.3 and relative error, SNR in Table 3.3, the sparse solution is relatively well restored, in spite of heavy noise data. In this experiments, we observe that the accelerated Bregman method overall performs better than the Bregman method and the model (3.45) work well when finding sparse solution.

3.4.2 Numerical results of inexact accelerated augmented Lagrangian method using various subproblem solvers

In this subsection, We consider the ℓ_1-ℓ_2 minimization problem with linear equality constraints :

$$\min_x \|x\|_1 + \frac{\beta}{2}\|x\|_2^2 \quad \text{such that} \quad Ax = b, \tag{3.46}$$

where $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^m$ and β is a positive constant.
This model (3.46) is same with (3.45) when $Q = \beta I$. In [15, 56], the authors proved the exact regularization property of ℓ_1-ℓ_2: the solution of (3.46) is also a solution of the basis pursuit if β is very small. In this test, we find the

sparse solution for solving linearly constrained ℓ_1-ℓ_2 minimization problem. With setting $f(x) = \|x\|_1 + \frac{\beta}{2}\|x\|_2^2$, the problem (3.46) is represented as the form (1.1), so we can apply ALM or AALM. However, the subproblem with respect to x in ALM or AALM

$$\arg\min_x \|x\|_1 + \frac{\beta}{2}\|x\|_2^2 - (\lambda_k)^T(Ax - b) + \frac{\tau}{2}\|Ax - b\|_2^2 \tag{3.47}$$

can not be solve exactly and the objective function $\|x\|_1 + \frac{\beta}{2}\|x\|_2^2$ is strongly convex function. Thus, we can apply the I-AALM for solving the linearly constrained ℓ_1-ℓ_2 minimization problem. For solving (3.47), we must apply other algorithm which solves unconstrained convex optimization. There are many algorithms for solving the unconstrained convex optimization as introduced in previous chapter. We explain some algorithms which can be applied the unconstrained convex optimization problem (2.9) and are some variants of ISTA.

Fixed Point method with BB line search(FP-BB) Under same setting in ISTA, we recall that the optimal solution of the problem (2.9) is a fixed point of the operator $(I + \tau\partial g)^{-1}(I - \tau\nabla f)$, for any $\tau > 0$. We rewrite the equivalence between the previous operator and quadratic approximation of $f + g$:

$$\begin{aligned} & x_{k+1} = (I + \tau\partial g)^{-1}(I - \tau\nabla f(x_k)) \\ \Leftrightarrow \quad & x_{k+1} = \arg\min_y f(x_k) + \langle y - x_k, \nabla f(x_k)\rangle + \frac{1}{2\tau}\|y - x_k\|_2^2 + g(y). \end{aligned}$$

Thus, ISTA is often called "Fixed Point method." In general, the Lipschitz constant L_f is not always easily computable, so, in order to find an approximated step size $\frac{1}{\tau}$, FP-BB use Barzilai-Borwein line search using Barzilai-Borwein steps [3].

Sparse Reconstruction by Separable Approximation(SpaRSA) We consider $F(x) = f(x) + \tau c(x)$ in (2.9), where f is a proper, smooth, convex function and c is finite and convex for all $x \in \mathbb{R}^n$. In SpaRSA, the subproblem

$$x_{k+1} = \arg\min_y f(x_k) + \langle y - x_k, \nabla f(x_k)\rangle + \frac{\alpha_k}{2}\|y - x_k\|_2^2 + \tau c(y)$$

is set up to solving problems of the form (2.9) with setting $F(x) = f(x) + \tau c(x)$. Actually, above problem is same with the step of ISTA by setting $g(x) = \tau c(x)$ and α_k instead of L_f. Hence, SpaRSA [52] is also closely related to ISTA. SpaRSA is an outer-inner iteration algorithm with respect to α_k. In inner iteration, the above subproblem is solved and then α_k is multiplied by positive constant $\eta > 1$ until the inner acceptance criterion is satisfied. In outer iteration, α_k is found using BB line search. The specification of SpaRSA is presented in the paper [52] and we can also find the convergence analysis of SpaRSA.

Now, we give the experimental results applying our proposed algorithm for solving linearly constrained ℓ_1-ℓ_2 minimization problem (3.46) and we use various algorithms, which are ISTA, FP-BB, FISTA and SpaRSA, for solving the subproblem (3.47). For experiments, we set $n = 500$ and $m = 250$ for the size of the measurement matrix A and we use the Gaussian measurement matrix A whose entries are randomly selected by standard Gaussian distribution and norm is 1, that is, $\|A^TA\|_2 = 1$. The ℓ_2 parameter β is fixed at 0.01 and the number k of nonzero elements of the original solution is fixed at 25. The locations of nonzero elements in the solution $\bar{x}$ are randomly chosen and the values of nonzero elements of $\bar{x}$ are chosen from standard Gaussian distribution. The penalty parameter γ is fixed to 100. For algorithms for solving subproblem (3.47) of I-A ALM, we use the decreasing sequence $\epsilon_k = \frac{10}{1.1^k}$. In this case, we can get the subdifferential of $f(x) = \|x\|_1 + \frac{\beta}{2}\|x\|_2^2$ as follows :

$$\partial f(x)_i = \begin{cases} \beta x_i + 1, & x_i > 0 \\ \beta x_i - 1, & x_i < 0 \\ \{y : -1 \le y \le 1\}, & x_i = 0 \end{cases},$$

where y_i means i-th element of y for any vector y. Hence, we can find the subgradient vector of f at x_{k+1} which is the closest vector to $A^T\lambda_{k+1} = A^T(\hat{\lambda}_k - \tau(Ax_{k+1} - b))$ for each iteration k and we can have the inexact stopping condition by simple calculation. In each test, we calculate the residual error $\|Ax_k - b\|_2$, the relative error

$$\frac{\|x_k - \bar{x}\|_2}{\|\bar{x}\|_2}$$

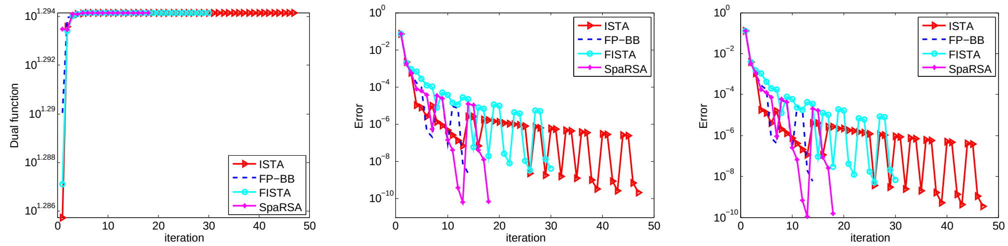

Figure 3.4: Results from ℓ_1-ℓ_2 minimization problem. Lagrangian dual function (1st column), relative error (2nd column) and residual error (3rd column) at each iteration

and the Lagrangian dual function at λ_k

$$D(\lambda_k) = -\frac{1}{2\beta}\|(|A^T\lambda_k| - 1)_+\|^2 + \lambda_k^T b$$

for each iteration k. We terminate the I-AALM when

$$\|Ax_k - b\| < 10^{-10}$$

is satisfied.

In Figure 3.4, we plot the Lagrangian dual function $D(\lambda_k)$, relative error and residual error at each iteration k. We observe that the Lagrangian dual function value monotone increases as the iteration number k increases. Although various algorithms use for solving subproblem (3.47) in I-AALM, Lagrangian dual function value in all tests converges same value. Since the I-AALM is not monotone method, the relative error and the residual error dose not decrease monotony, but both errors decay to zeros. Although same terminate condition uses in all tests, we observe that terminate iteration varies in all tests. Thus, we can show that the performance of I-AALM depends on the algorithms solving the subproblem (3.47).

3.4.3 Comparison to the inexact accelerated augmented Lagrangian method with other methods

In this subsection, we compare the performance of the I-AALM and state-of-the-arts algorithms for solving the linearly constrained ℓ_1-ℓ_2 minimization

(3.46). There are many algorithms for solving the ℓ_1-ℓ_2 minimization problem with linear equality constraints (3.46), which are original linearized Bregman method, accelerated schemes of linearized Bregman method. Yin et al. [56] proved that the linearized Bregman method is equivalent to a gradient descent method applied to the Lagrangian dual problem of (3.46). Based on this study, Yin [56] improved the linearized Bregman method using Barzilai-Borwein line search [3], the limited memory BFGS [29], and nonlinear conjugate gradient methods. Recently, the accelerated linearized Bregman method in [26] was developed based on Nesterov's acceleration of gradient method and Yang et al. [55] developed the linearized Bregman method-split Bregman method.

linearized Bregman method-split Bregman method(LB-SB) [55] To solving the linearly constrained ℓ_1-ℓ_2 minimization (3.46), the authors consider the dual formulation of (3.46):

$$\min_{y,z} -b^T y + \frac{1}{2\beta}\|A^T y - z\|_2^2 \quad \text{such that} \quad z \in [-1, 1], \tag{3.48}$$

with relation $x = \frac{1}{\beta} \cdot \text{shrink}(A^T y, 1)$ between primal and dual variables. Note that z can be expressed by $\text{Proj}_{[-1,1]}(A^T y)$. Based on this note, we easily verify that the dual problem (3.48) of (3.46) is equivalent to the unconstrained minimization problem

$$\min_{y} -b^T y + \frac{1}{2\beta}\|A^T y - \text{Proj}_{[-1,1]}(A^T y)\|_2^2. \tag{3.49}$$

By applying the variable splitting, the previous unconstrained optimization problem (3.49) is equivalent to the linearly constrained minimization problem

$$\min_{y,d} -b^T y + \frac{1}{2\beta}\|d - \text{Proj}_{[-1,1]}(d)\|_2^2 \quad \text{such that} \quad d = A^T y. \tag{3.50}$$

The problem (3.50) has the form (1.2), so, ADMM can be applied to this problem (3.50). Each subproblem with respect to y or d has the closed form solution. Based on this process, the dual split Bregman method is summarized in Algorithm 17.

We compare our I-AALM with state-of-the-art algorithms which are accelerated linearized Bregman method [26] and dual split Bregman method

Algorithm 17 LB-SB

Input : $\beta > 0, y_0 = \mathbf{0}, f_0 = \mathbf{0}$ and $\lambda = 10$

repeat

Update d_k following steps. $(\cdot)_i$ means i-th component.

if $|(A^T y_{k-1} + f_{k-1})_i| \leq 1$ **then**

$(d_k)_i = (A^T y_{k-1} + f_{k-1})_i$

else

$(d_k)_i = \dfrac{(A^T y_{k-1} + f_{k-1})_i + \text{sign}((A^T y_{k-1} + f_{k-1})_i)\lambda/\beta}{1 + \lambda/\beta}$

end if

$y_k = (AA^T)^{-1}(Ad_k - Af_{k-1} + \lambda b)$

$f_k = f_{k-1} + A^T y_k - d_k$

$x_k = \frac{1}{\beta} \cdot \text{shrink}(A^T y_k, 1)$

until *a stopping criterion is satisfied.*

[55] with the same setup for A and $\bar{x}$ as in the subsection 3.4.2 except for n, m, k and β. We set $m = 2500, n = 5000$ and sparsity $k = 250$ or $m = 1000, n = 5000$ and $k = 100$. We use various $\beta = 0.1, 0.01, 0.001$. We also give the observed vector the various noise :

$$b = A\bar{x} + n$$

where the noise n is generated by standard Gaussian distribution $\mathcal{N}(0, 1)$ and then it is normalized with the norm $\sigma = 0, 0.01$. When noise is present, we terminate all the methods when the residual error $\|Ax^k - b\|_2 \leq \sigma$ and for a noise-free case, we terminate all the methods when the relative residual error

$$\|Ax^k - b\|_2 < 5 \cdot 10^{-5} \|b\|_2$$

is satisfied. The coefficient γ in the penalty term of the subproblem of the I-AALM is fixed to $\gamma = 100$. We use the SpaRSA for solving the subproblem of the I-AALM. We also set a decreasing sequence

$$\epsilon_k = \begin{cases} \frac{1}{1.1^k}, & \beta = 0.1 \\ \frac{10}{1.1^k}, & \beta = 0.01 \\ \frac{100}{1.1^k}, & \beta = 0.001, \end{cases}$$

whose series $\sum_{k=1}^{\infty} \epsilon_k$ converges, for inexact stopping condition.

We record the average number of iterations, computing time, residual error, relative error and SNR for clean data in Tables 3.4-3.9, while we report results for noisy data with $\sigma = 0.01$ in Tables 3.10-3.15. In case of I-AALM, we report the number of total iteration which is summation of number of inner iterations.

As general conclusions we can say, the I-AALM is the fastest algorithm except for the one case that size of A is 2500×5000, $\beta = 0.1$ and $\sigma = 0$. We can see that the speed of I-AALM stays almost the same whenever we decrease the value of β for fixing noise level, size of A and sparsity. We can also observe that the recovery solution of I-AALM is well-restored for every case and it is slightly more accurate as we decrease the value of β. On the other hand, ALB get slower speed as we decrease β and the speed of ALB is less-affected by adding noise to b. The accuracy (relative error or SNR) of restored solution of ALB stays almost the same whenever we increase the value of β for noise-free case. The speed of LB-SB is very slow when we add noise to b and LB-SB has slightly slower speed as we decrease the value of β. When we give noise to b, LB-SB has less accurate than I-AALM or ALB. For all methods, as we increase number of measurement and density of $\bar{u}$, the speed is slower, although number of iterations stays almost the same. In conclusion, our proposed algorithm is the best algorithm in terms of accuracy and speed.

Let us comment on each single experiment in a little more details:

1. In first experiment (Table 3.4), all methods recover original sparse solution up to small relative error. Since I-AALM has inner iterations, we observe that the I-AALM has the smallest residual error and residual error of LB-SB is similar with that of ALB. Thus, the I-AALM has the smallest relative error and largest SNR and I-AALM find best-restored sparse solution. The I-AALM has the fastest runtime although I-AALM has larger number of iterations than ALB or LB-SB.

2. In second experiment (Table 3.5), we decrease the value of β. We observe that the performance of I-AALM is similar with that of I-AALM

for $\beta = 0.1$. Since average residual error is smaller than average residual error for $\beta = 0.1$, the average relative error is also smaller than that for $\beta = 0.1$ and average SNR is larger than that in the case of $\beta = 0.1$. ALB has very slower speed and the speed of LB-SB is slightly slower than the case of $\beta = 0.1$.

3. As we decrease very small value of β in Table 3.6, ALB has poor speed and many number of iterations, especially, I-AALM is around 12 times faster than ALB. The I-AALM is also around 2 times faster than LB-SB. From relative error or SNR in Table 3.6, we can show that the recovery solution of I-AALM has more accuracy than that of ALB or LB-SB.

4. By changing the size of A to 2500×5000 and having 250 nonzero entries of $\bar{x}$, we can see in 3.7 that all methods have slightly slower speed, especially, the LB-SB has larger computing time although average number of iterations for LB-SB is smaller than that in first experiment. We predict that speed of LB-SB is slow down because of computing inverse of the larger matrix. In only this case, the LB-SB is faster and has smaller number of iterations than I-AALM.

5. In this experiment (Table 3.8), we increase the value of β then, we can see that computing time of I-AALM and LB-SB stay almost the same in comparison with fourth experiment (Table 3.7) and the same is true for the number of iterations. However, I-AALM is slightly faster than LB-SB. Similar with second experiment (Table 3.5), average speed and number of iteration of ALB increase largely in comparison with fourth experiment.

6. By changing the value of β to smaller value 0.001, speed of I-AALM and LB-SB is similar with that at previous test, while number of iteration ALB is about 3 times larger than the case of $\beta = 0.01$. Average relative error and SNR of ALB and LB-SB stay almost the same whenever the value of β decreases. In the case of I-AALM, average relative error slightly larger than that for $\beta = 0.01$. On the contrary, average SNR is

about 122.252 up from 113.930 in $\beta = 0.01$. However, relative error or SNR is slightly different from that in $\beta = 0.01$.

7. By adding noise to b, the computation time and number of iterations of ALB and LB-SB grow so rapidly. Since stopping criterion is changed as adding noise to b, all algorithms can stop at larger residual error. For this reason, computing time and number of total iteration decrease a little bit in the case of I-AALM. Unusually, based on SNR or relative error in Table 3.10, we can observe that restored solution of LB-SB is very poor in comparison with that of ALB or I-AALM.

8. As value of β decreases, we can see that ALB has slower speed similar with the noise-free case and LB-SB has also slower speed, but this phenomenon is different from noise-free case. In particular, the computing time and number of iterations in LB-SB is larger than these in ALB with direct opposition to the case in Table 3.5. The speed of I-AALM is similar with that in the case of $\beta = 0.1$. average relative error and SNR increase in ALB and I-AALM, but decrease in LB-SB. I-AALM has still better restored-solution than that of ALB or LB-SB.

9. By modifying $\beta = 0.001$, LB-SB has very slow speed and relatively, LB-SB has less accurate restored solution based on SNR and relative error in Table 3.11. I-AALM has the 31 times faster speed than ALM and has as much as 141 times faster speed than LB-SB. ALB find more exact solution than the case of $\beta = 0.01$ and accuracy of restored solution of I-AALM remain almost like, whenever β is changed.

10. By modifying the size of A to 2500×5000 and having sparsity $k = 250$, computing time of LB-SB is very larger than that of ALB when $\beta = 0.1$. Additionally, LB-SB find less exact recovery solution than ALB or I-AALM. On the other hand, performance of I-AALM is the fastest and find the best accurate restored-solution.

11. Although we decrease the value of β, we can see a similar trend in Table 3.14. Continuously, LB-SB is very slow and has low accuracy restored solution when noise is added to b.

Table 3.4: Noise-free case. Size $A = 1000 \times 5000$, 2% of $\bar{u}$ is nonzero, $\beta = 0.1$

		mean	std.	maximum	minimum
Iter.	I-AALM	13.84(459.47)	7.60(224.89)	41(1292)	6(167)
(Total Iter.)	ALB	408.64	213.91	1084	174
	LB-SB	230.82	156.91	1114	155
Time	I-AALM	2.5234	1.2266	6.9829	0.9708
	ALB	3.0721	1.6158	8.4168	1.3081
	LB-SB	3.7019	2.4291	17.5928	2.5029
Res. Err.	I-AALM	5.860e-05	3.939e-05	1.529e-04	3.716e-06
	ALB	1.425e-04	1.575e-05	1.685e-04	6.473e-05
	LB-SB	1.399e-04	1.507e-05	1.709e-04	1.057e-04
Rel. Err.	I-AALM	2.288e-05	1.495e-05	5.822e-05	1.279e-06
	ALB	6.058e-05	5.898e-06	6.958e-05	2.898e-05
	LB-SB	5.998e-05	5.220e-06	6.830e-05	4.849e-05
SNR	I-AALM	95.488	7.944	117.862	84.698
	ALB	84.401	0.974	90.758	83.149
	LB-SB	84.473	0.775	86.285	83.310

12. When we set $\beta = 0.001$ and we add noise to b, average number of iterations in LB-SB has maximum value in all cases and that value is as much as 481 second. As we decrease the value of β, the restored solution of ALB or I-AALM is slightly more accurate, but it is only little different in essence. On the other hand, LB-SB find lower accurate solution than the case of $\beta = 0.1$. Thus, when we add noise to b, we can conclude that LB-SB may be unsuitable to use according to speed (computation time or number of iterations) and accuracy (SNR, relative error).

3.4.4 Inexact accelerated alternating direction method of multipliers for Multiplicative Noise Removal

The application of our I-AADMM is the problem of restoring a clean image from a noisy image corrupted by multiplicative noise. The multiplicative noise appears in ultrasound imaging, synthetic aperture radar (SAR) and sonar (SAS), laser imaging and magnetic field inhomogeneity in MRI. In

Table 3.5: Noise-free case. Size $A = 1000 \times 5000$, 2% of $\bar{u}$ is nonzero, $\beta = 0.01$

		mean	std.	maximum	minimum
Iter.	I-AALM	14.38(448.69)	7.04(204.75)	43(1119)	6(122)
(Total Iter.)	ALB	1301.36	679.18	4032	502
	LB-SB	277.17	151.61	1363	201
Time	I-AALM	2.4417	1.1178	6.2926	0.6877
	ALB	9.7400	5.2552	31.0395	3.5917
	LB-SB	4.4042	2.4997	22.9216	3.1630
Res. Err.	I-AALM	1.365e-05	1.940e-05	1.293e-04	1.077e-06
	ALB	1.443e-04	1.710e-05	1.908e-04	1.009e-04
	LB-SB	1.397e-04	1.610e-05	1.801e-04	1.058e-04
Rel. Err.	I-AALM	5.520e-06	7.662e-06	4.883e-05	3.676e-07
	ALB	6.040e-05	6.216e-06	7.027e-05	3.429e-05
	LB-SB	6.017e-05	5.247e-06	7.009e-05	4.223e-05
SNR	I-AALM	109.140	7.840	128.692	86.225
	ALB	84.431	0.996	89.296	83.061
	LB-SB	84.447	0.795	87.487	83.085

Table 3.6: Noise-free case. Size $A = 1000 \times 5000$, 2% of $\bar{u}$ is nonzero, $\beta = 0.001$

		mean	std.	maximum	minimum
Iter.	I-AALM	14.54(444.46)	7.83(213.90)	41(1272)	5(136)
(Total Iter.)	ALB	4107.48	2388.69	12124	1052
	LB-SB	326.35	167.68	1380	239
Time	I-AALM	2.3018	1.0966	6.5070	0.7336
	ALB	29.6453	17.2323	87.2290	7.5966
	LB-SB	4.9909	2.4808	20.5475	3.6606
Res. Err.	I-AALM	1.446e-05	2.660e-05	1.403e-04	6.993e-07
	ALB	1.421e-04	1.643e-05	1.885e-04	6.641e-05
	LB-SB	1.354e-04	2.025e-05	1.822e-04	7.099e-05
Rel. Err.	I-AALM	5.661e-06	9.903e-06	5.077e-05	2.928e-07
	ALB	6.043e-05	6.836e-06	6.983e-05	2.736e-05
	LB-SB	5.944e-05	8.227e-06	7.147e-05	3.268e-05
SNR	I-AALM	112.023	9.997	130.668	85.888
	ALB	84.444	1.176	91.259	83.120
	LB-SB	84.615	1.366	89.715	82.918

Table 3.7: Noise-free case. Size $A = 2500 \times 5000$, 5% of $\bar{u}$ is nonzero, $\beta = 0.1$

		mean	std.	maximum	minimum
Iter.	I-AALM	13.10(425.00)	5.18(157.51)	31(878)	5(163)
(Total Iter.)	ALB	344.78	144.30	798	135
	LB-SB	102.32	55.28	362	60
Time	I-AALM	5.4799	2.0281	11.2427	2.1229
	ALB	6.1618	2.5992	14.2068	2.4008
	LB-SB	5.2342	2.1821	15.5214	3.4063
Res. Err.	I-AALM	6.438e-05	7.084e-05	3.248e-04	1.204e-06
	ALB	2.898e-04	3.752e-05	3.670e-04	1.668e-04
	LB-SB	2.786e-04	3.928e-05	3.579e-04	1.801e-04
Rel. Err.	I-AALM	1.139e-05	1.209e-05	5.554e-05	2.250e-07
	ALB	5.646e-05	8.063e-06	6.790e-05	2.498e-05
	LB-SB	5.300e-05	6.860e-06	6.454e-05	3.497e-05
SNR	I-AALM	103.691	10.159	132.952	85.109
	ALB	85.068	1.413	92.046	83.362
	LB-SB	85.592	1.192	89.126	83.804

Table 3.8: Noise-free case. Size $A = 2500 \times 5000$, 5% of $\bar{u}$ is nonzero, $\beta = 0.01$

		mean	std.	maximum	minimum
Iter.	I-AALM	12.93(400.47)	5.12(160.40)	31(989)	6(151)
(Total Iter.)	ALB	1057.26	452.83	2620	405
	LB-SB	112.44	59.93	417	75
Time	I-AALM	5.1625	2.0480	12.6600	1.9516
	ALB	18.9107	8.1014	46.9173	7.2332
	LB-SB	5.6709	2.3591	17.7615	4.0957
Res. Err.	I-AALM	6.566e-05	9.876e-05	3.282e-04	6.093e-07
	ALB	2.836e-04	3.268e-05	3.600e-04	1.497e-04
	LB-SB	2.781e-04	3.561e-05	3.376e-04	1.959e-04
Rel. Err.	I-AALM	1.147e-05	1.715e-05	5.539e-05	1.015e-07
	ALB	5.578e-05	8.340e-06	6.924e-05	2.534e-05
	LB-SB	5.430e-05	6.678e-06	6.556e-05	3.899e-05
SNR	I-AALM	113.930	17.839	139.872	85.129
	ALB	85.186	1.504	91.924	83.188
	LB-SB	85.373	1.120	88.179	83.665

Table 3.9: Noise-free case. Size $A = 2500 \times 5000$, 5% of $\bar{u}$ is nonzero, $\beta = 0.001$

		mean	std.	maximum	minimum
Iter.	I-AALM	13.48(427.44)	4.76(155.23)	30(932)	6(116)
(Total Iter.)	ALB	3467.44	1357.63	8150	1351
	LB-SB	126.02	48.77	306	87
Time	I-AALM	5.6007	2.0104	12.2323	1.5594
	ALB	63.3687	24.8209	149.1493	24.6064
	LB-SB	6.3126	1.9639	13.4824	4.5769
Res. Err.	I-AALM	3.307e-05	6.819e-05	2.641e-04	2.928e-07
	ALB	2.889e-04	3.117e-05	3.419e-04	1.813e-04
	LB-SB	2.711e-04	4.122e-05	3.411e-04	1.558e-04
Rel. Err.	I-AALM	5.590e-06	1.126e-05	4.297e-05	4.954e-08
	ALB	5.672e-05	6.547e-06	6.750e-05	3.067e-05
	LB-SB	5.309e-05	7.585e-06	6.477e-05	3.100e-05
SNR	I-AALM	122.252	16.517	146.101	87.336
	ALB	84.987	1.081	90.265	83.414
	LB-SB	85.595	1.335	90.173	83.772

Table 3.10: Noise-added case. Size $A = 1000 \times 5000$, 2% of $\bar{u}$ is nonzero, $\beta = 0.1$

		mean	std.	maximum	minimum
Iter.	I-AALM	7.41(247.68)	1.38(90.44)	11(570)	4.00(61)
(Total Iter.)	ALB	660.06	372.72	1017	123
	LB-SB	517.80	39.37	600	386
Time	I-AALM	1.2858	0.4493	2.8669	0.3417
	ALB	6.3292	3.5855	10.0232	1.1564
	LB-SB	7.8709	0.5961	9.1464	5.8928
Res. Err.	I-AALM	9.542e-03	2.418e-04	9.988e-03	9.109e-03
	ALB	9.841e-03	1.536e-04	1.000e-02	9.232e-03
	LB-SB	9.824e-03	1.315e-04	9.997e-03	9.383e-03
Rel. Err.	I-AALM	1.730e-03	3.967e-04	3.227e-03	9.305e-04
	ALB	4.524e-03	2.274e-03	7.402e-03	1.237e-03
	LB-SB	7.028e-03	6.152e-04	8.546e-03	5.737e-03
SNR	I-AALM	55.460	1.977	60.624	49.824
	ALB	48.413	5.585	58.150	42.613
	LB-SB	43.095	0.767	44.826	41.362

Table 3.11: Noise-added case. Size $A = 1000 \times 5000$, 2% of $\bar{u}$ is nonzero, $\beta = 0.01$

		mean	std.	maximum	minimum
Iter.	I-AALM	7.39(231.29)	1.25(76.79)	11(478)	5(69)
(Total Iter.)	ALB	1513.83	1398.41	3914	374
	LB-SB	2626.77	189.63	3135	2129
Time	I-AALM	1.2307	0.3857	2.4620	0.4018
	ALB	14.7370	13.6041	38.0016	3.6205
	LB-SB	40.0628	2.8931	47.4138	32.2973
Res. Err.	I-AALM	9.561e-03	2.047e-04	9.988e-03	9.078e-03
	ALB	9.902e-03	9.618e-05	9.996e-03	9.452e-03
	LB-SB	9.933e-03	5.105e-05	9.999e-03	9.728e-03
Rel. Err.	I-AALM	1.558e-03	3.149e-04	2.608e-03	1.031e-03
	ALB	2.945e-03	2.041e-03	7.600e-03	1.228e-03
	LB-SB	7.182e-03	5.451e-04	8.618e-03	5.859e-03
SNR	I-AALM	56.313	1.702	59.735	51.674
	ALB	52.352	5.212	58.216	42.378
	LB-SB	42.899	0.656	44.642	41.291

Table 3.12: Noise-added case. Size $A = 1000 \times 5000$, 2% of $\bar{u}$ is nonzero, $\beta = 0.001$

		mean	std.	maximum	minimum
Iter.	I-AALM	7.63(222.66)	1.58(78.76)	11(486)	4(69)
(Total Iter.)	ALB	3816.13	4180.94	14358	943
	LB-SB	11093.84	1270.58	14763	8843
Time	I-AALM	1.1583	0.3861	2.4329	0.3806
	ALB	36.2291	39.7064	136.4345	8.8944
	LB-SB	164.1395	18.8453	219.6760	131.1316
Res. Err.	I-AALM	9.630e-03	2.147e-04	9.997e-03	9.073e-03
	ALB	9.935e-03	8.443e-05	1.000e-02	9.460e-03
	LB-SB	9.963e-03	2.553e-05	9.999e-03	9.873e-03
Rel. Err.	I-AALM	1.634e-03	3.549e-04	2.780e-03	9.244e-04
	ALB	2.312e-03	1.686e-03	7.366e-03	1.053e-03
	LB-SB	7.057e-03	6.328e-04	8.917e-03	5.657e-03
SNR	I-AALM	55.943	1.936	60.682	51.118
	ALB	54.132	4.373	59.553	42.655
	LB-SB	43.061	0.778	44.948	40.991

Table 3.13: Noise-added case. Size $A = 2500 \times 5000$, 5% of $\bar{u}$ is nonzero, $\beta = 0.1$

		mean	std.	maximum	minimum
Iter.	I-AALM	7.47(235.35)	1.14(73.50)	10(494)	5(89)
(Total Iter.)	ALB	299.79	375.11	1733	116
	LB-SB	509.34	33.75	577	416
Time	I-AALM	3.0294	0.9005	6.1662	1.2402
	ALB	7.1908	9.0235	41.8323	2.7654
	LB-SB	21.1852	1.3550	23.8966	17.6403
Res. Err.	I-AALM	9.653e-03	1.901e-04	9.989e-03	9.377e-03
	ALB	9.912e-03	8.181e-05	1.000e-02	9.612e-03
	LB-SB	9.797e-03	1.516e-04	1.000e-02	9.410e-03
Rel. Err.	I-AALM	6.709e-04	9.152e-05	9.307e-04	4.969e-04
	ALB	9.403e-04	6.491e-04	3.427e-03	5.622e-04
	LB-SB	3.645e-03	1.911e-04	4.366e-03	3.234e-03
SNR	I-AALM	63.546	1.175	66.074	60.624
	ALB	61.485	3.350	65.002	49.300
	LB-SB	48.777	0.452	49.804	47.198

Table 3.14: Noise-added case. Size $A = 2500 \times 5000$, 5% of $\bar{u}$ is nonzero, $\beta = 0.01$

		mean	std.	maximum	minimum
Iter.	I-AALM	7.64(234.78)	1.17(79.93)	11(424)	5(81)
(Total Iter.)	ALB	636.45	127.46	935	329
	LB-SB	2717.18	206.89	3276	2266
Time	I-AALM	3.0461	0.9884	5.3953	1.1021
	ALB	15.4051	3.0955	22.4608	8.0013
	LB-SB	108.6316	8.2255	131.1447	90.9103
Res. Err.	I-AALM	9.690e-03	1.815e-04	9.997e-03	9.407e-03
	ALB	9.919e-03	9.309e-05	1.000e-02	9.495e-03
	LB-SB	9.932e-03	5.510e-05	9.999e-03	9.757e-03
Rel. Err.	I-AALM	6.691e-04	9.167e-05	9.350e-04	4.712e-04
	ALB	7.482e-04	7.182e-05	8.920e-04	5.904e-04
	LB-SB	3.936e-03	2.053e-04	4.447e-03	3.501e-03
SNR	I-AALM	63.571	1.204	66.536	60.583
	ALB	62.559	0.852	64.577	60.990
	LB-SB	48.110	0.451	49.114	47.037

Table 3.15: Noise-added case. Size $A = 2500 \times 5000$, 5% of $\bar{u}$ is nonzero, $\beta = 0.001$

		mean	std.	maximum	minimum
Iter.	I-AALM	7.45(218.26)	1.29(80.99)	11(477)	5(81)
(Total. Iter.)	ALB	1961.25	386.61	2885	1127
	LB-SB	11961.34	1099.85	15039	8781
Time	I-AALM	2.8866	1.0137	5.8436	1.1520
	ALB	48.2929	9.5364	71.2886	27.8140
	LB-SB	481.5565	44.0275	604.6174	354.1902
Res. Err.	I-AALM	9.656e-03	1.713e-04	9.996e-03	9.312e-03
	ALB	9.914e-03	8.128e-05	1.000e-02	9.612e-03
	LB-SB	9.964e-03	2.902e-05	1.000e-02	9.880e-03
Rel. Err.	I-AALM	6.452e-04	8.550e-05	8.311e-04	4.779e-04
	ALB	7.424e-04	6.239e-05	8.831e-04	5.665e-04
	LB-SB	3.923e-03	1.799e-04	4.316e-03	3.448e-03
SNR	I-AALM	63.880	1.137	66.414	61.606
	ALB	62.617	0.736	64.935	61.079
	LB-SB	48.136	0.401	49.248	47.299

this work, we assume the degradation model as

$$f = g \cdot n, \tag{3.51}$$

where $f : \Omega \to \mathbb{R}$ is an observed noisy data defined on an open and bounded set $\Omega \subset \mathbb{R}^2$, g is the ideal image to be recovered, and n is the noise that follows a Gamma distribution with $\mathbb{E}(n) = 1$, commonly occuring in SAR (known as speckle).

In variational approaches, several work devoted to multiplicative noise removal have been proposed, such as Rudin et al. [45], Aubert and Aujol [2], Shi and Osher [49], Huang et al. [27], Steidl and Teuber [50], etc. As a seminal work, Aubert and Aujol [2] used a maximum a posteriori (MAP) regularization approach and derived a functional whose minimizer corresponds to the denoised image to be recovered. This functional is

$$\min_g \alpha \int_\Omega \left(\log g + \frac{f}{g} \right) \, dx + \int_\Omega |\nabla g| \, dx, \tag{3.52}$$

where the total variation of g is utilized as the regularization term, and

$\alpha > 0$ is the parameter that controls the smoothness of the restored image g. Despite the noncovexity of the functional, the authors proved the existence of a minimizer, and provided a sufficient condition for the uniqueness.

Huang et al. [27] proposed a strictly convex TV minimization function for multiplicative noise removal. The authors used logarithmic transformation on both side of (3.51) and converted the multiplicative problem into the additive one: $\log f = \log g + \log n$. Then they added a quadratic term in the data term of the Aubert and Aujol model (3.52) and replaced the regularizer for g by z by using an auxiliary variable $z = \log g$. Therefore, their proposed model is described as

$$\min_{z,u} \alpha \int_\Omega (z + fe^{-z})\ dx + \frac{\mu}{2} \int_\Omega (z-u)^2\ dx + \int_\Omega |\nabla u|\ dx, \tag{3.53}$$

where $\mu > 0$ is the parameter that measures the trade-off between an image obtained by a maximum likelihood estimation from the first term and a total variation denoised image u. The main advantage of this model is that the total variation regularization enables us to preserve edges well in the denoised image.

It is known that the total variation denoising method preserves edges well but produces undesirable staircasing effect in the denoised image, since it favor piecewise constant solutions. To ameliorate the staircasing effect, a popular approach is the use of the higher-order regularization. The most of the higher-order noms involve second-order differential operators because second-order derivatives lead to piecewise-linear solutions that better fit smooth intensity changes. We propose a hybrid total variation minimization model for multiplicative noise removal, so that it reduces the staircasing effect while preserving the discontinuities (edges) as well as that our I-AADMM is applicable. Therefore, the proposed model is as follows

$$\begin{aligned}\min_{z,u} E(z,u) = \alpha \int_\Omega (z + fe^{-z})\ dx + \frac{\beta}{2} \int_\Omega z^2\ dx + \frac{\mu}{2} \int_\Omega (z-u)^2\ dx \\ + \int_\Omega |\nabla u| + \frac{\epsilon}{2}|\nabla u|^2\ dx + \frac{\delta}{2} \int_\Omega |\nabla^2 u|^2\ dx,\end{aligned} \tag{3.54}$$

where $\epsilon, \delta > 0$ are parameters.

Now we introduce auxiliary variables d and w, and convert the proposed unconstrained minimization problem (3.54) to the constrained one as

$$\begin{aligned}\min_{z,u,d,w} F(z,u,d,w) &= \alpha \int (z + fe^{-z})\ dx + \frac{\beta}{2}\int z^2\ dx + \frac{\mu}{2}\int (z-u)^2\ dx \\ &\quad + \int |d| + \frac{\epsilon}{2}|d|^2\ dx + \frac{\delta}{2}\|w\|_2^2, \\ \text{s.t.}\quad d &= \nabla u, \quad w = \nabla^2 u. \end{aligned} \tag{3.55}$$

This constrained model can be rewritten as

$$\min_{z,u,d,w} F(z,u,d,w) = H(z,u,d) + G(w), \quad \text{s.t.} \quad B\begin{pmatrix} z \\ u \\ d \end{pmatrix} + Cw = \mathbf{0}, \tag{3.56}$$

with the functionals H, G and the matrices B, C defined as

$$\begin{aligned} H(z,u,d) &= \alpha\int z + fe^{-z}\ dx + \frac{\beta}{2}\int z^2\ dx + \frac{\mu}{2}\int (z-u)^2\ dx \\ &\quad + \int |d| + \frac{\epsilon}{2}|d|^2\ dx, \\ G(w) &= \frac{\delta}{2}\|w\|_2^2, \\ B &= \begin{pmatrix} 0 & -\nabla & I \\ 0 & -\nabla^2 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 \\ I \end{pmatrix}, \end{aligned}$$

where $\rho(B^TB) = \|\Delta\|_2^2 = 64$ and $\rho(C^TC) = 1$.

It is trivial to show that the functional H is strongly convex with respect to z, with modulus β. Hence, H is strongly convex with respect to (z,u,d) with the modulus

$$\sigma_H = \min(\lambda_{\min}(Hess), \epsilon), \tag{3.57}$$

where $Hess = \begin{pmatrix} \beta+\mu & -\mu \\ -\mu & \mu \end{pmatrix}$ and $\lambda_{\min}(Hess)$ is the minimum eigenvalue of the matrix $Hess$. Moreover, G is also strongly convex with the modulus $\sigma_G = \delta$, and it is quadratic.

Therefore, the I-AADMM algorithm to solve the problem (3.56) is given by

$$\begin{aligned}
(z_k, u_k, d_k) &\approx \arg\min_{z,u,d} \Big\{ H(z,u,d) - \hat{\lambda}_k^T \cdot \begin{pmatrix} -\nabla u + d \\ -\nabla^2 u \end{pmatrix} + \frac{\tau}{2}\|d - \nabla u\|_2^2 \\
&\qquad + \frac{\tau}{2}\|\hat{w}_k - \nabla^2 u\|_2^2 \Big\}, \\
w_k &\approx \arg\min_{w} \Big\{ G(w) - \hat{\lambda}_k^T \cdot \begin{pmatrix} 0 \\ w \end{pmatrix} + \frac{\tau}{2}\|d_k - \nabla u_k\|_2^2 + \frac{\tau}{2}\|w - \nabla^2 u_k\|_2^2 \Big\}, \\
\lambda_k &= \hat{\lambda}_k - \tau \begin{pmatrix} d_k - \nabla u_k \\ w_k - \nabla^2 u_k \end{pmatrix}, \qquad (3.58) \\
\alpha_{k+1} &= \frac{1 + \sqrt{1 + 4\alpha_k^2}}{2}, \\
\hat{w}_{k+1} &= w_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(w_k - w_{k-1}), \\
\hat{\lambda}_{k+1} &= \lambda_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\lambda_k - \lambda_{k-1}),
\end{aligned}$$

where $\hat{\lambda}_k^T = (\hat{\lambda}_{1,k}, \hat{\lambda}_{2,k})$.

To solve the subproblem for (z_k, u_k, d_k), we take the partial derivatives of the energy with respect to z, u, d, leading to the normal equations as below

$$\alpha(1 - fe^{-z}) + \beta z + \mu(z - u) = 0, \qquad (3.59)$$

$$(\mu - \tau\Delta + \tau\Delta^2)u = \mu z - \tau\nabla \cdot (d - \frac{\hat{\lambda}_{1,k}}{\tau}) + \tau \mathrm{div}^2(\hat{w}_k - \frac{\hat{\lambda}_{2,k}}{\tau}), \qquad (3.60)$$

$$d = shrink(\frac{\tau\nabla u + \hat{\lambda}_{1,k}}{\tau + \epsilon}, \frac{1}{\epsilon + \tau}). \qquad (3.61)$$

To overcome the expensive computation of inverse of coefficient matrix, we use the semi-implicit scheme, which alternatively solves the Euler-Lagrange equations. Fixing u and d, we begin with solving the first equation for z_{k_n}, which can be very efficiently determined by using the Newton method. On the other hand, when z and d are fixed, the solution u_{k_n} can be obtained by one-step FFT implementation by assuming the periodic boundary condition. Lastly, the iterates d_{k_n} can be obtained with the one-step thresholding operator. We iterate this process for $k_n = 0, 1, ...$ until the stopping criteria is reached, to obtain (z_k, u_k, d_k).

Finally, the minimizing solution w_k can be obtained with the explicit formula

$$w_k = \frac{\tau(\nabla^2 u_k) + \hat{\lambda}_{2,k}}{\delta + \tau}. \tag{3.62}$$

Hence, the summary of our iterative algorithm is as follows:

- For (z_k, u_k, d_k), iterate for $k_n = 0, 1, ...$ until the stopping criteria, with $(z_{k_0}, u_{k_0}, d_{k_0}) = (z_{k-1}, u_{k-1}, d_{k-1})$:
  - For z_{k_n}: iterate for $\ell = 0, 1, ..., L$ (in practice, we set $L = 5$) with $z_{k_n,0} = z_{k_{n-1}}$,

$$z_{k_n,\ell+1} = z_{k_n,\ell} - \frac{\alpha(1 - fe^{-z_{k_n,\ell}}) + \beta z_{k_n,\ell} + \mu(z_{k_n,\ell} - u_{k_{n-1}})}{\alpha f e^{-z_{k_n,\ell}} + \beta + \mu}$$

  - For u_{k_n}: perform one-step FFT implementation

$$u_{k_n} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mu z_{k_n} - \tau\nabla\cdot(d_{k_{n-1}} - \frac{\hat{\lambda}_{1,k}}{\tau}) + \tau\mathrm{div}^2(\hat{w}_k - \frac{\hat{\lambda}_{2,k}}{\tau}))}{\mu - \gamma\mathcal{F}(\Delta) + \gamma\mathcal{F}(\Delta)^2}\right)$$

  - For d_{k_n}:

$$d_{k_n} = shrink\Big(\frac{\tau\nabla u_{k_n} + \hat{\lambda}_{1,k}}{\tau + \epsilon}, \frac{1}{\epsilon + \tau}\Big)$$

- For w_k:

$$w_k = \frac{\tau(\nabla^2 u_k) + \hat{\lambda}_{2,k}}{\delta + \tau}$$

- Update:

$$\begin{aligned}
\lambda_{1,k} &= \hat{\lambda}_{1,k} - \tau(d_k - \nabla u_k), \\
\lambda_{2,k} &= \hat{\lambda}_{2,k} - \tau(w_k - \nabla^2 u_k), \\
\alpha_{k+1} &= \frac{1 + \sqrt{1 + 4\alpha_k^2}}{2}, \\
\hat{w}_{k+1} &= w_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(w_k - w_{k-1}), \\
\hat{\lambda}_{1,k+1} &= \lambda_{1,k} + \frac{\alpha_k - 1}{\alpha_{k+1}}(\lambda_{1,k} - \lambda_{1,k-1}), \\
\hat{\lambda}_{2,k+1} &= \lambda_{2,k} + \frac{\alpha_k - 1}{\alpha_{k+1}}(\lambda_{1,k} - \lambda_{1,k-1}).
\end{aligned}$$

For inexact stopping conditions in I-AADMM, we set the sequence $\epsilon_k = \frac{10^5}{1.1^k}$ for all $k \geq 1$. Although we take a large ϵ_0, the series $\sum_{k=0} \epsilon_k$ is converges. Hence, we take $\hat{v}_1 = \mathbf{0}$. For solving TV model (3.53), we apply alternating minimization algorithm which do the minimization for z, u in an alternating fashion.

$$\begin{cases} z_{k+1} = \min_z \alpha \int_\Omega \left(z + fe^{-z}\right) \, dx + \frac{\mu}{2} \int_\Omega (z - u_k)^2 \, dx & (3.63a) \\ u_{k+1} = \min_u \frac{\mu}{2} \int_\Omega (z_{k+1} - u)^2 \, dx + \int_\Omega |\nabla u| \, dx, & (3.63b) \end{cases}$$

To solve the equation for z, we also use Newton method as our model. To solve the second equation for u, we apply one-step split Bregman method [19].

$$\min_{u,d} \frac{\mu}{2} \int_\Omega (z_{k+1} - u)^2 \, dx + \int_\Omega |d| \, dx, \quad \text{s.t.} \quad d = \nabla u.$$

by variable splitting. Then, using alternating direction method of multipliers, we obtain

$$\begin{aligned} u_{k+1,n} &= \arg\min_u \left\{ \frac{\mu}{2} \int_\Omega (z_{k+1} - u)^2 \, dx + \lambda_{n-1}^T (\nabla u) + \frac{\tau}{2} \|d_n - \nabla u\|_2^2 \right\} \\ d_n &= \arg\min_d \left\{ \int_\Omega |d| \, dx - \lambda_{n-1}^T (d) + \frac{\tau}{2} \|d - \nabla u_{k+1,n}\|_2^2 \right\} \\ \lambda_n &= \lambda_{n-1} - \tau (d_n - \nabla u_{k+1,n}) \end{aligned}$$

and we take $n = 1$ for fast speed, i.e. we perform only one ADMM iteration. The summary of alternating minimization algorithm for TV model is as follows:

- For z_k: iterate for $\ell = 0, 1, ..., L$ (in practice, we set $L = 5$) with $z_{k,0} = z_{k-1}$,

$$z_{k,\ell+1} = z_{k,\ell} - \frac{\alpha(1 - fe^{-z_{k,\ell}}) + \mu(z_{k,\ell} - u_{k-1})}{\alpha f e^{-z_{k,\ell}} + \mu}$$

- For u_k: perform one-step FFT implementation

$$u_k = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(\mu z_k - \tau \nabla \cdot (d_{k-1} - \frac{\lambda_{k-1}}{\tau}))}{\mu - \gamma \mathcal{F}(\Delta)} \right)$$

Table 3.16: Comparison results of TV model and Our model

	SNR		Total Iteration		Time(s)	
	Our	TV	Our	TV	Our	TV
Circle	28.36	27.41	1851	704	38.22	12.29
Satellite	16.04	15.61	2105	502	173.06	32.38
Barbara	19.08	17.88	2358	728	44.07	11.90

- For d_k:

$$d_k = shrink\Big(\frac{\tau\nabla u_k + \lambda_k}{\tau}, \frac{1}{\tau}\Big)$$

- Update of dual variables λ_k:

$$\lambda_k = \lambda_{k-1} - \tau(d_k - \nabla u_k).$$

Then, the outer iteration for two models (TV model (3.53) and our model (3.54)) is stopped when

$$\frac{|E(u^k, z^k) - E(u^{k-1}, z^{k-1})|}{|E(u^k, z^k)|} < tol,$$

where tol is a threshold defining the desired accuracy. As default value we use $tol = 10^{-4}$.

In Figures 3.5-3.7, the data f are corrupted by the speckle noise following a Gamma density with $\mathbb{E}(n) = 1$ and $\sigma_n^2 = 1/L$. First, we can observe that our hybrid model combined with the second-order regularization ameliorates staircasing effect arisen from the total variation term. This leads to more natural looking restored images while preserving details better than the TV model. This shows that our additional quadratic second order regularization term enhances the denoising results in the presence of high density of multiplicative noise. The SNR values in Table 3.16 also yield that denoised image of our model is better than that of TV model. We can also show convergence of both algorithms from the graphs of log scale of residual error. In Table 3.16, we observed that the I-AADMM for our model has slower speed than the alternating minimization algorithm for the TV model. However, convergence of our algorithm was proved in the previous section. On the other

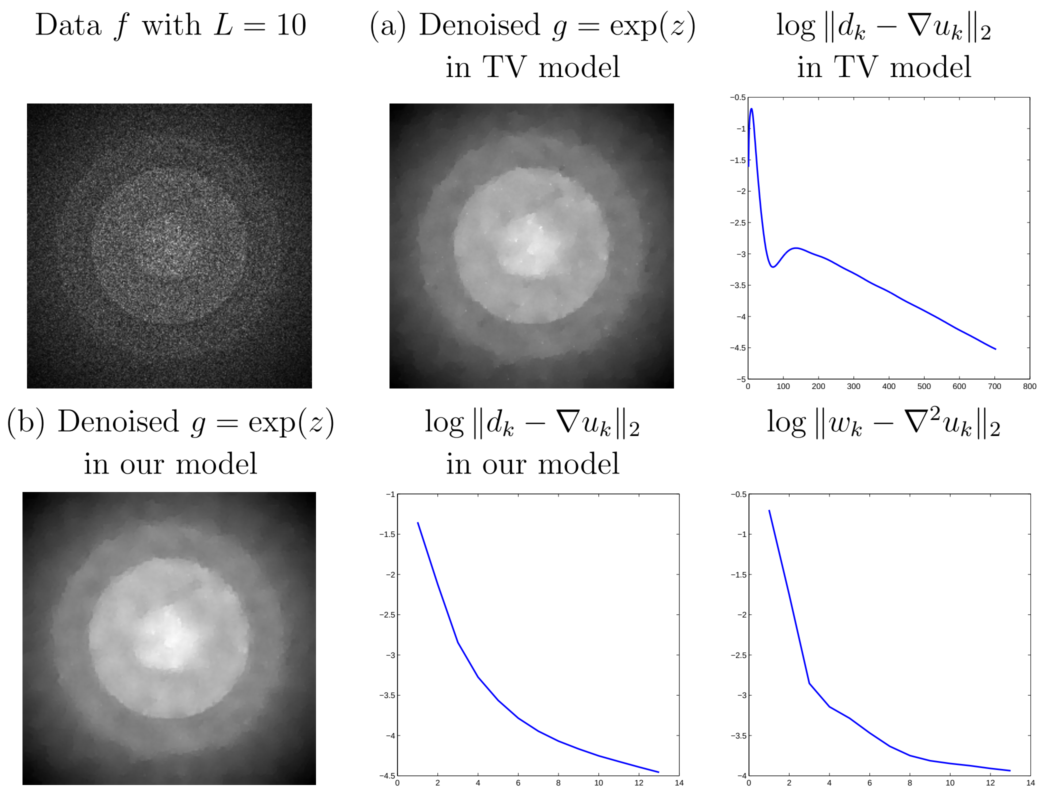

Figure 3.5: Denoising results of (b) our model and comparison with (a) the TV model (3.53). Parameter: $\alpha = 3$, $\beta = 0.005$ (ours), $\alpha = 3$ (TV), $(\epsilon, \delta, \mu, \beta) = (0.01, 1, 500, 0.005)$.

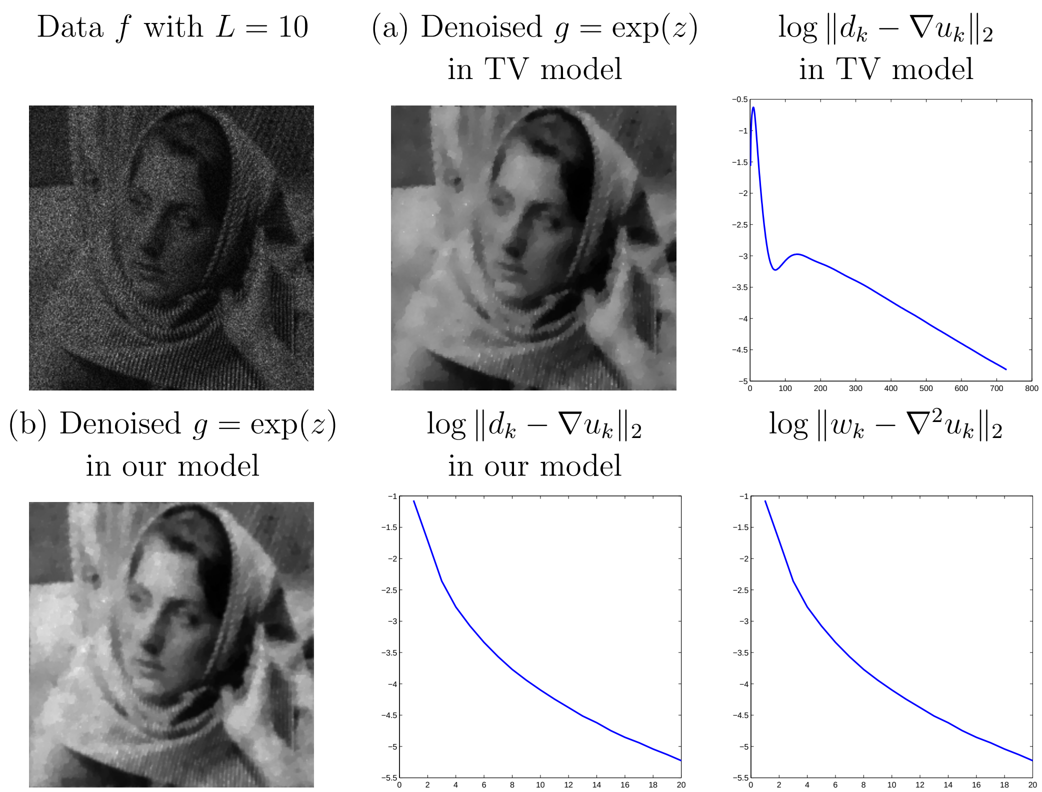

Figure 3.6: Denoising results of (b) our model and comparison with (a) the TV model (3.53). Parameter: $\alpha = 4$ (ours), $\alpha = 3$ (TV), $(\epsilon, \delta, \mu, \beta) = (0.01, 1, 500, 0.005)$.

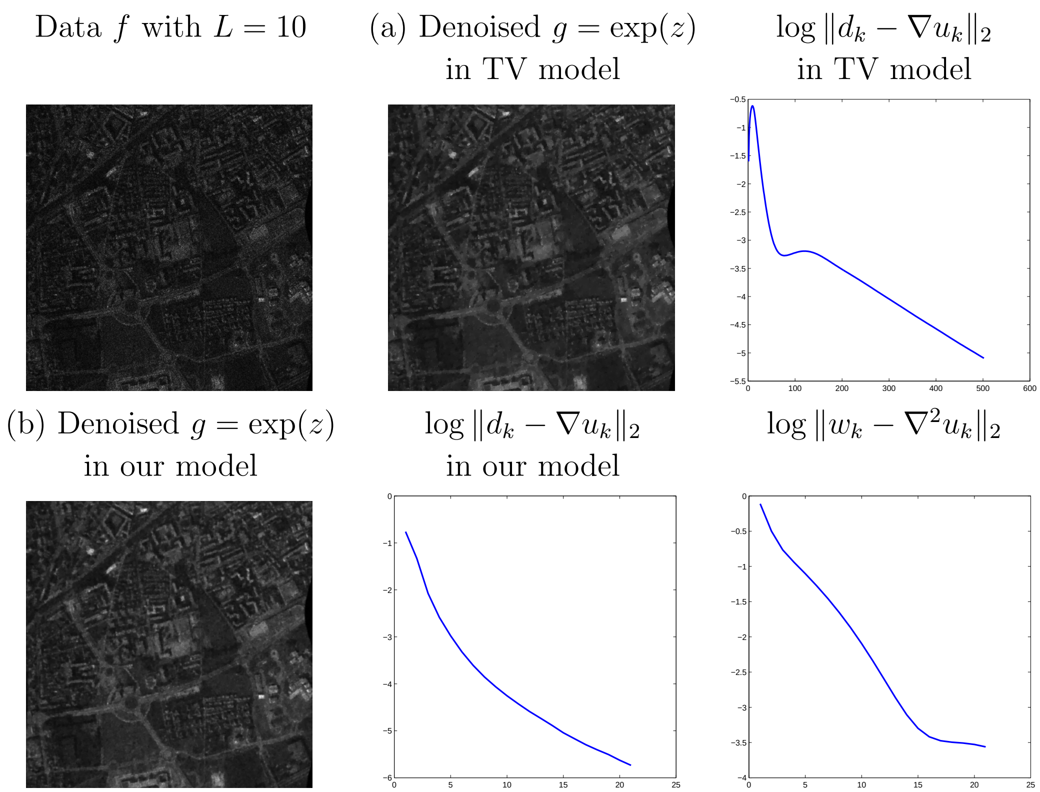

Figure 3.7: Denoising results of (b) our model and comparison with (a) the TV model (3.53). Parameter: $\alpha = 9$, $\beta = 0.005$ (ours), $\alpha = 5$ (TV), $(\epsilon, \delta, \mu, \beta) = (0.01, 1, 500, 0.005)$.

hand, for the subproblem of the alternating minimization algorithm, we just use one iteration of the ADMM. Actually, we apply the alternating split Bregman method to the TV model which has coupled objective function. When the alternating split Bregman method is applied to problem with coupled objective function, its convergence has not been proved, although each subproblem can be solved exactly. In conclusion, our algorithm is slower than the alternating minimization algorithm for the TV model in numerical tests, but the convergence of our algorithm is proven theoretically while the alternating minimization algorithm for the TV model converges in numerical tests without theoretical analysis.

Chapter 4

Conclusion

We proposed the new algorithms for solving linearly constrained convex minimization problems (1.1) and (1.2) in this dissertation.

The first method is an accelerated algorithm for the Bregman method. We have shown that the convergence rate of the original Bregman method is $\mathcal{O}(\frac{1}{k})$ and that of the accelerated Bregman method is $\mathcal{O}(\frac{1}{k^2})$ for general linearly constrained nonsmooth convex minimization, based on the equivalence between the Bregman method and the augmented Lagrangian method. According to numerical test, we showed that the proposed algorithm is faster than the original Bregman method when we solve the linearly constrained ℓ_1 and generalized ℓ_2 minimization.

The first method is an inexact version of the accelerated augmented Lagrangian method (AALM). Despite acceleration of the convergence rate, the computational cost of the accelerated methods is comparable with that of the original ones. This is mainly due to the subproblem minimization required to be solved exactly at each (outer) iteration. In general, the subproblem in the ALM does not have a closed-form solution. Therefore, we have developed an inexact version of the AALM (I-AALM), with an stopping condition for the subproblem. It is also proven that the convergence rate of the I-AALM remains the same as the AALM. The numerical results related to the linearly constrained ℓ_1-ℓ_2 minimization problem show that the proposed I-AALM has outstanding speed and accuracy of recovery, compared with state-of-art algorithms. The proposed method can only apply the problem whose object function is a strongly convex function and the objective function of models in many applications is not a strongly convex function. In future research, we will develop the inexact conditions, which are computed numerically, of subproblem in (accelerated) augmented Lagrangian method when object function is an convex, proper and lower semicontinuous function.

The last is the I-AADMM, which is an inexact version of the fast alternating direction method of multipliers (FADMM). Again, inexact stopping criterions of the subproblems are provided, and it is proven that the convergence rate is $\mathcal{O}(\frac{1}{k^2})$ under the same conditions with the FADMM. As an application, we introduced a new variational model for multiplicative noise removal, which incorporates the total variation regularization with a higher-order one. When heavy noise is given, our model with the proposed algorithm provides better denoised images than the model with the total variation (TV) regularizer, visually and according to the SNR values. Even though the total speed of our algorithm is slower than the alternating minimization algorithm applied to the TV model, our algorithm is more based on a theoretical analysis. The alternating minimization algorithm is the other popular method for solving (1.2). This algorithm is the same with alternating direction method of multipliers except for first subproblem. If we use the similar technique with the case of I-AADMM, we will most likely be able to develop the inexact version of accelerated alternating minimization algorithm in future.

Bibliography

[1] Y. I. Alber, R. S. Burachik, and A. N. Iusem, *A proximal point method for nonsmooth convex optimization problems in Banach spaces*, Abstr. Appl. Anal. **2** (1997), 97–120.

[2] G. Aubert and J.-F. Aujol, *A variational approach to removing multiplicative noise*, SIAM J. Applied Mathematics **68** (2008), no. 4, 925–946.

[3] J. Barzilai and J. Borwein, *Two point step size gradient methods*, IMA J. Numer. Anal. **8** (1988), 141–148.

[4] A. Beck and M. Teboulle, *A fast iterative shrinkage-threshoilding algorithm for linear inverse problems*, SIAM J. Imag. Sci. **2** (2009), no. 1, 183–202.

[5] A. Benfenati and V. Ruggiero, *Inexact Bregman iteration with an application to poisson data reconstruction*, Inverse Problems **29** (2013), no. 6, 065016.

[6] D. P. Bertsekas, *Constrained optimization and lagrange mupltiplier methods*, Athena Scientific (1996).

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learning **3** (2010), no. 1, 1–122.

[8] J. Cai, S. Osher, and Z. Shen, *Linearized Bregman iterations for compressed sensing*, Mathematics of Comput. **78** (2009), 1515–1536.

[9] E. Candés, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inf. Thoery **52** (2006), 489–509.

[10] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, *An introduction to total variation for image analysis*, in Theoretical Foundations and Numerical Methods for Sparse Recovery, M. Fornasier, ed., Radon Ser. Comput. Appl. Math., vol. 9, Walter de Gruyter, Berlin, 2010.

[11] F. Donga, H. Zhangb, and D.-X. Kongc, *Nonlocal total variation models for multiplicative noise removal using split bregman iteration*, Mathematical and Computer Modelling (2012), 939–954.

[12] J. Douglas and H. Rachford, *On the numerical solution of heat conduction problems in two and three space variables*, Transactions of the American Mathematical Society **82** (1956), no. 3, 421–439.

[13] M. Elad, B. Matalon, and M. Zibulevsky, *Subspace optimization methods for linear least squares with non-quadratic regularization*, Appl. Comput. Harmon. Anal. **23** (2007), 346–367.

[14] E. Esser, *Applications of Lagrangian-based alternating direction methods and connections to split-Bregman*, UCLA CAM Report 09–31 (2009).

[15] J. Freiedlander and P. Tseng, *Exact regularization of convex programs*, SIAM J. Opt. **18** (2007), 1326–1350.

[16] O. Güler, *New proximal point algorithms for convex minimization*, SIAM J. on Optimization **2** (1992), no. 4, 649–664.

[17] P. Güler, *On the convergence of the proximal point algorithm for convex minimization*, **29** (1991), no. 2, 403–419.

[18] T. Goldstein, B. O'Donoghue, and S. Setzer, *Fast alternating direction optimization methods*, UCLA CAM Report 12-35 (2012).

[19] T. Goldstein and S. Osher, *The split bregman method for l1-regularized problems*, SIAM J. Imag. Sci. **2** (2009), no. 2, 323–343.

[20] E. Hale, W. Yin, and Y. Zhang, *Fixed-point continuation applied to compressed sensing: Implementation and numerical experiments*, J. Comput. Math. **28** (2010), 170–194.

[21] B. He and H. Yang, *Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities*, Oper. Res. Lett. **23** (1998), 151–161.

[22] B. He and X. Yuan, *The acceleration of augmented lagrangian method for linearly constrained optimization*, optimization online www.optimization-online.org/DB FILE/2010/10/2760.pdf (2010).

[23] ———, *An accelerated inexact proximal point algorithm for convex minimization*, J. Optim. thoery Appl. **154** (2012), no. 2.

[24] M. R. Hestenes, *Multiplier and gradient methods*, J. Optim. thoery Appl. **4** (1969), 303–320.

[25] J.-B. Hiriart-Urruty and C. Lemarechal, *Convex analysis and minimization algorithms: Part 1:fundamentals*, Springer-Verlag, New York, 1996.

[26] B. Huang, S. Ma, and D. Goldfarb, *Accelerated linearized Bregman method*, J. Sci. Comput. **54** (2013), 428–453.

[27] Y.-M. Huang, M. K. Ng, and Y.-W. Wen, *A new total variation method for multiplicative noise removal*, SIAM J. Imag. Sci. **2** (2009), no. 1, 20–40.

[28] K. Jiang, D. Sun, and K-C Toh., *An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP*, SIAM J. Optim. **22** (2012), no. 3, 1042–1064.

[29] D. Liu and J. Nocedal, *On the limited memory method for large scale optimization*, Math.Programming Ser. B **45** (1989), 503–528.

[30] B. Martinet, *Régularisation d'inéquations variationalles par approximations sucessives*, Rev. Française Informat. Recherche Opérationnelle **4** (1970), 154–158.

[31] J. J. Moreau, *Proximité et dualité dans un espace hilbertien*, Bull. Soc. Math. France **93** (1965), 273–299.

[32] V. Nedelcu and I. Necoara, *Iteration complexity of an inexact augmented lagrangian method for constrained mpc*, IEEE 51st Annual conference on Decision and Control (2012), 650–655.

[33] A. S. Nemirovsky and D. B. Yudin, *Problem complexity and method efficiency in optimization*, John Wiley and Sons Ltd, 1983.

[34] Y. Nesterov, *A method of solving a convex programming problem with convergence rate $\mathcal{O}\left(\frac{1}{k^2}\right)$*, In Soviet Mathematices Doklady **27** (1983), 372–376.

[35] ———, *An approach to constructing optimal methods for minimization of smooth convex functions*, Ékonom. i Mat. Metody **24** (1988), no. 3, 97–116.

[36] ———, *Introductory lectures on convex optimization. a basic course*, Kluwer Academic Publishers, Boston, 2004.

[37] ———, *Gradient methods for minimizing composite objective function*, CORE (2007).

[38] M. K. Ng, F. Wang, and X. Yuan, *Inexact alternating direction methods for image recovery*, SIAM J. Sci. Comput. **33** (2011), no. 4, 1643–1668.

[39] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, SIAM J. Multiscale Model. and Simul. **4** (2005), 460–489.

[40] S. Osher, Y. Mao, B. Dong, and W. Yin, *Fast linearized Bregman iteration for compressive sensing and sparse denoising*, Comm. in Math. Sciences. **8** (2010), 93–111.

[41] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, Academic Press (1972), 283–298.

[42] R. Rockafellar, *Augmented lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res. **1** (1976), no. 2, 97–116.

[43] ———, *Monotone operators and the proximal point algorithm*, **14** (1976), no. 5, 877–898.

[44] R. Rockafellar and R. J.-B. Wets, *Variational analysis*, Springer-Verlag, New York, 1998.

[45] L. Rudin, P.-L. Lions, and S. Osher, *Multplicative denoising and deblurring: theory and algorithms*, in Geometric Level Sets in Imaging, Vision and Graphics, O. Osher and N. Paragios, Eds., Springer, New York, NY, USA (2003), 103–119.

[46] L. Rudin, S.Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D. **60** (1992), 259–268.

[47] S. Salzo and S. Villa, *Inexact and accelerated proximal point algorithms*, J. Convex Anal. **19** (2012).

[48] S. Setzer, *Operator splittings, Bregman methods and frame shrinkage in image processing*, Int. J. Comput. Vis. **92** (2011), 265–280.

[49] J. Shi and S. Osher, *A nonlinear inverse scale space method for a convex multiplicative noise model*, SIAM J. Imag. Sci. **1** (2008), no. 3, 294–321.

[50] G. Steidl and T. Teuber, *Removing multiplicative noise by douglas-rachford splitting methods*, Journal of Mathematical Imaging and Vision **36** (2010), no. 2, 168–184.

[51] S. Villa, S. Salzo, L. BalDassarre, and A. Verri, *Accelerated and inexact forward-backward algorithms*, SIAM J. Optim. **23** (2013), no. 3, 1607–1633.

[52] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, *Sparse reconstruction by separable approximation*, IEEE Trans. Signal Proc. **57** (2009), 2479–2493.

[53] J. Yang and Y. Zhang, *Alternating direction algorithms for ℓ_1-problems in compressive sensing*, SIAM J. Sci. Comput. **33**, no. 1, 250–278.

[54] J. Yang, Y. Zhang, , and W. Yin, *A fast alternating direction method for tvl1-l2 signal reconstruction from partial fourier data*, IEEE J. Selected Topics in Signal Processing **4** (2010), no. 2, 288–297.

[55] Y. Yang, M. Móller, and S. Osher, *A dual split Bregman method for fast ℓ_1 minimization*, Mathematics of Computation **82** (2013), no. 284, 2061–2085.

[56] W. Yin, *Analysis and generalizations of the linearized bregman method*, SIAM J. Imag. Sci. **31** (2010), 856–877.

[57] W. Yin and S. Osher, *Error forgetting of Bregman iteration*, J. Sci. Comput. **54** (2013), 684–695.

[58] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for ℓ_1-minimization with applications to compressive sensing*, SIAM J. Imag. Sci. **1** (2008), 143–168.

국문초록

선형 제한 조건의 수학적 최적화는 다양한 영상 처리 문제의 모델로서 사용되고 있다. 이 논문에서는 이 선형 제한 조건의 수학적 최적화 문제를 풀기위한 빠른 알고리듬들을 소개하고자 한다. 우리가 제안하는 방법들은 공통적으로 Nesterov에 의해서 개발되었던 가속화한 프록시말 그레디언트 방법에서 사용되었던 보외법을 기초로 하고 있다. 여기에서 우리는 크게보아서 두가지 알고리듬을 제안하고자 한다. 첫번째 방법은 가속화한 Bregman 방법이며, 압축센싱문제에 적용하여서 원래의 Bregman 방법보다 가속화한 방법이 더 빠름을 확인한다. 두번째 방법은 가속화한 어그먼티드 라그랑지안 방법을 확장한 것인데, 어그먼티드 라그랑지안 방법은 내부문제를 가지고 있고, 이런 내부문제는 일반적으로 정확한 답을 계산할 수 없다. 그렇기 때문에 이런 내부문제를 적당한 조건을 만족하도록 부정확하게 풀더라도 가속화한 어그먼티드 라그랑지 방법이 정확하게 내부문제를 풀때와 같은 수렴성을 갖는 조건을 제시한다. 우리는 또한 가속화한 얼터네이팅 디렉션 방법데 대해서도 비슷한 내용을 전개한다.

주요어휘: 어그먼티드 라그랑지안 방법, Bregman 방법, 압축센싱 ,Nesterov의 가속화 방법 ,최적화 이론
학번: 2008-20274