CLC__________ Number__________

UDC__________ Available for reference ☐Yes ☐No

Undergraduate Thesis

On ADMM for Three Separable Operators and Accelerated Algorithms

Student Name : Li Yao

Student ID : 11310025

Department : Mathematics

Program : Mathematics and Applied Mathematics

Thesis Advisor : Prof. Bingsheng He

Date : April, 2017

COMMITMENT OF HONESTY

1. I solemnly promise that the paper presented comes from my independent research work under my supervisor's supervision. All statistics and images are real and reliable.
2. Except for the annotated reference, the paper contents no other published work or achievement by person or group. All people making important contributions to the study of the paper have been indicated clearly in the paper.
3. I promise that I did not plagiarize other people's research achievement or forge related data in the process of designing topic and research content.
4. If there is violation of any intellectual property right, I will take legal responsibility myself.

Signature:

Date:

摘 要

乘子交替方向法（ADMM）是用来处理具有可分离结构的凸优化问题的算法。这个方法通过交替迭代优化两个子问题从而简化了原始问题的求解难度。本文研究了 ADMM 在目标函数为三个可分离算子情形下的改进算法。对直接推广 ADMM 不收敛的情况，基于变分不等式和预测 -校正算法框架给出两种校正算法。对于第二种校正算法，进一步研究了其算法结构，给出了一个衍生的校正算法。最后利用两个简单的算例进行验证。本文共分为五章：

第一章为绪论和预备知识，引出三个可分离算子情形的 ADMM 问题，并且简要介绍了相关的基本概念以及本文的符号约定和假设。

第二章引入基于变分不等式的算法预测 -校正算法框架，为第三章的算法收敛性提供理论基础。

第三章给出了两种校正算法，并在第二章的预测 -校正算法框架下完成了收敛性的证明。进一步研究了第二种校正算法的结构，给出了其变形形式。

第四章进一步拓展了第二种校正算法，通过加入松弛因子使得算法可以以更大的步长迭代。

第五章利用 matlab 编程给出了对直接推广 ADMM 算法不收敛的例子, 对含有 elastic net 正则项的 Lasso 问题和对有约的 Lasso 问题的算法实现。

关键词： 凸优化，ADMM，可分离结构，变分不等式，松弛因子

ABSTRACT

The alternating direction method of multipliers(ADMM) is an algorithm which is used to solve convex optimization problems with separable structure. It simplifies the difficulty of original problem by optimizing two subproblems alternately and iteratively. In this paper, improved algorithms of ADMM for three separable operators are concerned with. Two improved algorithms based on variational inequality and prediction-correction framework are provided in the case that the straightforward extension of ADMM for three separable operators may not converge. For the second improved algorithm, further study on its algorithm structure generates a variant of it. In the end, two numerical examples verify the effectiveness and efficiency of algorithms. The dissertation embraces five chapters:

Chapter 1 is about introduction and preliminary knowledge. It introduces ADMM for three separable operators. Furthermore, some relevant basic concepts, assumptions and notations are provided.

Chapter 2 introduces the prediction-correction algorithm framework based on variational inequality which is the foundation of convergence proof in Chapter 3.

Chapter 3 covers two improved algorithms and convergence analysis of them in the framework introduced in Chapter 2.

Chapter 4 further extends the second improved algorithm by adding a relaxation factor to iterate with larger step.

Chapter 5 deals with three numerical results, the counterexample which is not convergent in direct extension of ADMM, Lasso with elastic net and Lasso with constraints, by using Matlab.

Keywords: convex optimization, ADMM, separable structure, variational inequality, relaxation factor

Contents

List of Figures

List of Tables

List of Matlab Codes

Chapter 1 Introduction

1.1 Preliminaries

Formally, there are two parts in this section. We refer to [1, 2] to give some basic definitions and theorems of convex optimization in the first part and variational inequality [3, 4] will be included in the second part, which would contribute to the understanding of the following chapters.

Definition 1.1.1. *A set $\Omega \subset \Re^n$ is convex if it satisfies*

$$\alpha x + (1-\alpha)y \in \Omega, \quad \forall x, y \in \Omega, \alpha \in [0,1]$$

.

Definition 1.1.2. *Let $\Omega \subset \Re^n$ be convex, then a function $f : \Omega \to \Re$ is convex if it satisfies*

$$f(\alpha x + (1-\alpha)y) \le \alpha f(x) + (1-\alpha) f(y), \quad \forall x, y \in \Omega, \alpha \in [0,1]$$

.

In practical problems, we can always transform optimization problems into the follow unifying forms:

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \le 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{aligned} \tag{1.1.1}$$

where $f_0 : \Re^n \to \Re$ is called objective function, the inequalities $f_i : \Re^n \to \Re$, $i = 1, \ldots, m$ are called inequality constraints and the equalities $h_i : \Re^n \to \Re$, $i = 1, \ldots, p$ are called equality constraints. We call $\mathcal{D} = \{x \in \Re^n | f_i(x) \le 0,\ i = 1, \ldots, m,\ h_i(x) = 0,\ i = 1, \ldots, p\}$ is feasible region of problem (1.1.1).

In this paper, we only focus on convex optimization problems which have convex objective function and convex feasible region. Note that, convex region require that inequality constraint functions are convex and equality constraint functions are affine, which makes (1.1.1) transformed into

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \le 0, \quad i = 1, \ldots, m \\ & a_i^T x = b_i, \quad i = 1, \ldots, p, \end{aligned} \tag{1.1.2}$$

where f_0 is convex objective function, f_i, $i = 1, \ldots, m$ is convex inequality constraint function and $a_i \in \Re^n, b_i \in \Re$, $i = 1, \ldots, p$.

For optimization problem (1.1.1), assume feasible region $\mathcal{D}$ is not empty, we can define its Lagrange function $L : \Re^n \times \Re^m \times \Re^p \to \Re$ as

$$L(x, \lambda, \nu) := f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x), \tag{1.1.3}$$

where λ_i, ν_i are Lagrange multipliers and λ, ν are Lagrange multiplier vectors.

Definition 1.1.3. *For $\lambda \in \Re^m$, $\nu \in \Re^p$, the function*

$$g(\lambda,\nu) := \inf_{x\in\mathcal{D}} L(x,\lambda,\nu) = \inf_{x\in\mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right)$$

is called Lagrange dual function(or dual function).

We use p^* to denote the optimal value of problem (1.1.1), then it is not difficult to observe that $g(\lambda,\nu) \le p^*$ for arbitrary $\lambda \succeq 0$($\lambda \succeq 0$ means each component is no less than 0) and ν. Under this circumstance, $g(\lambda,\nu)$ gives a lower bound of optimal value p^*. It is natural to think what is the best lower bound generated by dual function, which is expressed as

$$\begin{array}{ll} \text{maximize} & g(\lambda,\nu) \\ \text{subject to} & \lambda \succeq 0. \end{array} \tag{1.1.4}$$

(1.1.4) is called Lagrange dual problem, and we call (λ^*,ν^*) an optimal Lagrange multiplier if it is optimal solution of (1.1.4). We use d^* to denote the optimal value of dual problem, then it is obvious that

$$d^* \le p^*. \tag{1.1.5}$$

This property is called weak duality.

If the equality

$$d^* = p^* \tag{1.1.6}$$

is established, then it is called strong duality. We can also use saddle point to give an explanation of strong duality, that is

$$L(x^*,\lambda,\nu) \le L(x^*,\lambda^*,\nu^*) \le L(x,\lambda^*,\nu^*), \tag{1.1.7}$$

where x^* is optimal point of original problem(1.1.1) or (1.1.2), (λ^*,ν^*) is optimal point of dual problem(1.1.4) and the triplet of variables (x^*,λ^*,ν^*) is a saddle point of Lagrange function.

The next theorem indicates (1.1.7) are sufficient conditions for any optimization problems.

Theorem 1.1.1. *Consider the optimization problem given by (1.1.1) where f_0 and f_i and g_i are arbitrary functions. If a triplet of variables (x^*,λ^*,ν^*) exists for the Lagrange function(1.1.3), satisfying (1.1.7) for all $x \in \mathcal{D}$, $\lambda \succeq 0$ and $\nu \in \Re^p$. x^* must be an optimal solution to the optimization problem(1.1.1).*

For general optimization problems, (1.1.6) and (1.1.7) are not established. However, convex optimization problems(1.1.2) may establish them by satisfying some conditions. Meanwhile, KKT saddle point conditions are necessary for any convex optimization problems.

In this paper, we only focus on convex optimization problems with linear equality constraints,

$$\begin{array}{ll} \text{minimize} & \theta(x) \\ \text{subject to} & Ax = b \end{array} \tag{1.1.8}$$

where $x \in \Omega \subset \Re^n$, $A \in \Re^{m\times n}$, $b \in \Re^m$ and θ is convex and Ω is closed, convex.

For optimization problem without constraints like

$$\theta(x) + f(x), \tag{1.1.9}$$

where $x \in \mathcal{X}$, $\mathcal{X}$ is closed, convex subset of $\Re^n$, θ,f are convex functions and f is differentiable. Assume solution set of (1.1.8) is not empty, we can reformulate it into variational inequality by following lemma.

Lemma 1.1.1.

$$x^* := \arg\min_{x\in\mathcal{X}} \left\{\theta(x) + f(x)\right\}, \tag{1.1.10}$$

if and only if

$$x^* \in \mathcal{X}, \quad \theta(x) - \theta(x^*) + (x - x^*)^T \nabla f(x^*) \geq 0, \quad \forall x \in \mathcal{X}. \tag{1.1.11}$$

Proof. Firstly, if (1.1.10) is established, then for each $x \in \mathcal{X}$, define

$$x_n := \frac{1}{n}x + (1 - \frac{1}{n})x^*, \quad n \geq 1,$$

we have

$$\theta(x_n) - \theta(x^*) + f(x_n) - f(x^*) \geq 0. \tag{1.1.12}$$

Since

$$\theta(x_n) \leq \frac{1}{n}\theta(x) + (1 - \frac{1}{n})\theta(x^*), \tag{1.1.13}$$

then

$$\theta(x) - \theta(x^*) \geq \frac{\theta(x_n) - \theta(x^*)}{\frac{1}{n}}, \tag{1.1.14}$$

and substituting (1.1.14) into (1.1.12), we have

$$\theta(x) - \theta(x^*) + \frac{f(x_n) - f(x^*)}{\frac{1}{n}} \geq 0. \tag{1.1.15}$$

Since

$$\frac{f(x_n) - f(x^*)}{\frac{1}{n}} = \frac{f(x^* + \frac{1}{n}(x - x^*)) - f(x^*)}{\frac{1}{n}}, \tag{1.1.16}$$

let $n \to \infty$, we have

$$\theta(x) - \theta(x^*) + (x - x^*)^T \nabla f(x^*) \geq 0. \tag{1.1.17}$$

Conversely, since f is convex, it also has

$$f(x_n) - f(x^*) \leq \frac{1}{n}(f(x) - f(x^*)), \tag{1.1.18}$$

then for each $n \geq 1$,we have

$$f(x) - f(x^*) \geq \frac{f(x_n) - f(x^*)}{\frac{1}{n}}. \tag{1.1.19}$$

Let $n \to \infty$, then

$$f(x) - f(x^*) \geq \nabla f(x^*)^T (x - x^*) \geq 0. \tag{1.1.20}$$

Substituting (1.1.20) into (1.1.11), we have

$$x^* \in \mathcal{X}, \quad \theta(x) - \theta x^* + f(x) - f(x^*) \geq 0, \quad \forall x \in \mathcal{X}, \tag{1.1.21}$$

and the proof is complete. □

The next proposition is frequently used in the following chapters.

Proposition 1.1.1. *Assume H is a nonnegative definite matrix, then there exists identity*

$$(a-b)^T H(c-d) = \frac{1}{2}\|a-d\|_H^2 - \frac{1}{2}\|a-c\|_H^2 + \frac{1}{2}\|b-c\|_H^2 - \frac{1}{2}\|b-d\|_H^2, \quad (1.1.22)$$

where $\|\cdot\|_H^2 = (\cdot)^T H(\cdot)$.

1.2 Backgrounds

ADMM(Alternating Direction Method of Multipliers) is an efficient method to solve large-scale, linearly constrained convex optimization problem with separable structure. Although it is applicable in many areas nowadays, such as in engineering design, statistical learning and graphical models, it is a method that develops in the 1970s [5, 6], with the roots in 1950s.

The problem solved by ADMM is stated as

$$\begin{array}{ll} \text{minimize} & \theta_1(x) + \theta_2(y) \\ \text{subject to} & Ax + By = b \end{array} \quad (1.2.1)$$

where $x \in \mathcal{X} \subset \Re^{n_1}$, $y \in \mathcal{Y} \subset \Re^{n_2}$, $A \in \Re^{m\times n_1}$, $B \in \Re^{m\times n_2}$, $b \in \Re^m$ and θ_1, θ_2 are convex and $\mathcal{X}, \mathcal{Y}$ are closed, convex.

Compared to linearly constrained convex optimization problem (1.1.8), the variable of (1.2.1) can be separated into two parts. The objective function and linear constraint also have corresponding separation. We call this kind of structure is separable.

For problem (1.1.8), there is an algorithm called ALM(Augmented Lagrangian Method),

$$\begin{cases} x^{k+1} = \arg\min\{L_\beta(x, \lambda^k) | x \in \Omega\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} - b), \end{cases} \quad (1.2.2)$$

where $L_\beta(x,\lambda) = \theta(x) - \lambda^T(Ax-b) + \frac{\beta}{2}\|Ax-b\|^2$ and $\beta > 0$.

We call $L_\beta(x,\lambda)$ with $\beta > 0$ as augmented Lagrangian function and it should be noticed that augmented Lagrangian function just adds a quadratic penalty function to Lagrangian function to bring robustness. The algorithm (1.2.2) is well-studied in [7, 8] and it is also related to other algorithms like PPA(proximal point algorithm). ALM is applicable to problem (1.2.1) but the separable structure is not used. Thus, ADMM is like a mixture of ALM and a decomposition method which iterates in Gauss-Seidal manner to solve (1.2.1),

$$\begin{cases} x^{k+1} = \arg\min\{L_\beta(x, y^k, \lambda^k) | x \in \mathcal{X}\}, \\ y^{k+1} = \arg\min\{L_\beta(x^{k+1}, y, \lambda^k) | y \in \mathcal{Y}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b), \end{cases} \quad (1.2.3)$$

where $L_\beta(x,y,\lambda) = \theta_1(x) + \theta_2(y) - \lambda^T(Ax+By-b) + \frac{\beta}{2}\|Ax+By-b\|^2$.

Algorithm (1.2.3) is convergent and related to some other algorithms, such as Douglas-Rachford splitting from numerical analysis, Spingarn's method of partial inverses, Dykstra's alternating projections method, Bregman iterative algorithms for l_1 problems in signal processing, proximal methods, and many others.

Some accelerated algorithms of ADMM are also studied, such as ADMM with self-adaptive penalty parameter[9], ADMM with relaxation factor[10] and so on.

With the wide application of ADMM in distributed optimization, it is natural idea to extend ADMM for three-operator problem, that is,

$$\begin{array}{ll} \text{minimize} & \theta_1(x) + \theta_2(y) + \theta_3(z) \\ \text{subject to} & Ax + By + Cz = b \end{array} \tag{1.2.4}$$

where $x \in \mathcal{X} \subset \Re^{n_1}$, $y \in \mathcal{Y} \subset \Re^{n_2}$, $z \in \mathcal{Z} \subset \Re^{n_3}$, $A \in \Re^{m\times n_1}$, $B \in \Re^{m\times n_2}$, $C \in \Re^{m\times n_3}$, $b \in \Re^m$ and $\theta_1, \theta_2, \theta_3$ are convex and $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ are closed, convex.

The direct extension method of ADMM for problem (1.2.4) is stated as

$$\begin{cases} x^{k+1} = \arg\min\{L_\beta(x, y^k, z^k, \lambda^k) | x \in \mathcal{X}\}, \\ y^{k+1} = \arg\min\{L_\beta(x^{k+1}, y, z^k, \lambda^k) | y \in \mathcal{Y}\}, \\ z^{k+1} = \arg\min\{L_\beta(x^{k+1}, y^{k+1}, z, \lambda^k) | z \in \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b). \end{cases} \tag{1.2.5}$$

For a long time, the convergent property of algorithm (1.2.5) is not clear until there is a counterexample to be given in [11]. Even we assume objective function is strongly convex, the model-independent parameter β cannot be found. In this thesis, two improved algorithms and an accelerated algorithm are given to solve problem (1.2.4).

In the following chapters, we use w to denote (x, y, z, λ), u to denote (x, y, z) and v to denote (y, z, λ) if not specifically stated. Since we don't focus on calculation of the x, y and z-subproblems, we assume these three subproblems are not difficult to solve and the solution set of problem(1.2.4) is not empty. For simplicity, we also assume coefficient matrix A, B and C all have full column rank.

Chapter 2 Prediction-Correction Framework

In this chapter, we will use variational inequality to give prediction-correction algorithm framework and prove the convergence and convergence rate. Firstly, we reformulate the optimal conditions into the form of variational inequality.

2.1 Optimal Condition

By **Theorem 1.1.1**, assume $(x^*, y^*, z^*) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ is optimal point of three separable operators problem(1.2.4), then saddle inequalities exist, which is,

$$L_{\lambda \in \Re^m}(x^*, y^*, z^*, \lambda) \leq L(x^*, y^*, z^*, \lambda^*) \leq L_{(x,y,z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}}(x, y, z, \lambda^*), \tag{2.1.1}$$

where $\lambda^* \in \Re^m$ is optimal point of Lagrangian dual function.

The inequality (2.1.1) can be rewritten as

$$\begin{cases} x^* \in \mathcal{X}, & L(x, y^*, z^*, \lambda) - L(x^*, y^*, z^*, \lambda^*) \geq 0, \quad \forall x \in \mathcal{X}, \\ y^* \in \mathcal{Y}, & L(x^*, y, z^*, \lambda) - L(x^*, y^*, z^*, \lambda^*) \geq 0, \quad \forall y \in \mathcal{Y}, \\ z^* \in \mathcal{Z}, & L(x^*, y^*, z, \lambda) - L(x^*, y^*, z^*, \lambda^*) \geq 0, \quad \forall z \in \mathcal{Z}, \\ \lambda^* \in \Re^m, & L(x^*, y^*, z^*, \lambda^*) - L(x^*, y^*, z^*, \lambda) \geq 0, \quad \forall \lambda \in \Re^m, \end{cases} \tag{2.1.2}$$

that is,

$$\begin{cases} x^* & = \arg\min\{L(x, y^*, z^*, \lambda^*) | x \in \mathcal{X}\}, \\ y^* & = \arg\min\{L(x^*, y, z^*, \lambda^*) | y \in \mathcal{Y}\}, \\ z^* & = \arg\min\{L(x^*, y^*, z, \lambda^*) | z \in \mathcal{Z}\}, \\ \lambda^* & = \arg\max\{L(x^*, y^*, z^*, \lambda) | \lambda \in \Re^m\}. \end{cases} \tag{2.1.3}$$

By lemma 1.1.1, we expand (2.1.3) and get

$$\begin{cases} x^* \in \mathcal{X}, & \theta_1(x) - \theta_1(x^*) + (x - x^*)^T(-A^T\lambda^*) \geq 0, \quad \forall x \in \mathcal{X}, \\ y^* \in \mathcal{Y}, & \theta_2(y) - \theta_2(y^*) + (y - y^*)^T(-B^T\lambda^*) \geq 0, \quad \forall y \in \mathcal{Y}, \\ z^* \in \mathcal{Z}, & \theta_3(z) - \theta_3(z^*) + (z - z^*)^T(-C^T\lambda^*) \geq 0, \quad \forall z \in \mathcal{Z}, \\ \lambda^* \in \Re^m, & (\lambda - \lambda^*)^T(Ax^* + By^* + Cz^* - b) \geq 0, \quad \forall \lambda \in \Re^m. \end{cases} \tag{2.1.4}$$

We can reformulate (2.1.4) into a compact form,

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall w \in \Omega, \tag{2.1.5}$$

where

$$F(w) = \begin{pmatrix} -A^T\lambda \\ -B^T\lambda \\ -C^T\lambda \\ Ax + By + Cz - b \end{pmatrix},$$

$$\theta(u) = \theta_1(x) + \theta_2(y) + \theta_3(z), \quad \text{and} \quad \Omega = \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \times \Re^m.$$

Remark 2.1.1. *It should be noticed that $F(w)$ is skew-symmetric which satisfies*

$$(w - \bar{w})^T(F(w) - F(\bar{w})) = 0, \quad \forall w, \bar{w} \in \Omega. \tag{2.1.6}$$

2.2 Description of Framework

Under the prediction-correction algorithm framework introduced in this section, we reform the problem in form of variational inequality (2.1.5), that is,

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall w \in \Omega.$$

Denote $v = (y, z, \lambda) \in \mathcal{Y} \times \mathcal{Z} \times \Re^m$, then we can describe the framework in following two steps.

Prediction: we use a provided v^k to generate a $\widetilde{w}^k$ such that

$$\widetilde{w}^k \in \Omega, \quad \theta(u) - \theta(u^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \geq (v - \widetilde{v}^k)^T Q(v^k - \widetilde{v}^k), \quad \forall w \in \Omega. \tag{2.2.1}$$

Correction: the new iterative point v^{k+1} is generated by

$$v^{k+1} = v^k - M(v^k - \widetilde{v}^k). \tag{2.2.2}$$

In this prediction-correction algorithm framework, the convergence analysis of ADMM and improved algorithms of three separable operators problem(1.2.4) is simple and easy to understand.

2.3 Convergence Analysis

Lemma 2.3.1. *Assume $Q^T + Q$ in (2.2.1) is positive definite, and for matrix M in (2.2.2), there exists a positive definite matrix H such that*

$$HM = Q, \tag{2.3.1}$$

and

$$G = Q^T + Q - M^T HM \succ 0, \quad \textit{(at least positive semi-definite)}, \tag{2.3.2}$$

then the iterative point generated by prediction-correction algorithm satisfies

$$\begin{aligned} \widetilde{w}^k \in \Omega, \quad & \theta(u) - \theta(\widetilde{u}^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \\ & \geq \frac{1}{2}(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2) + \frac{1}{2}\|v^k - \widetilde{v}^k\|_G^2, \quad \forall w \in \Omega. \end{aligned} \tag{2.3.3}$$

Proof. By (2.2.2), since $Q = HM$, (2.2.1) can be rewritten as

$$\begin{aligned} \widetilde{w}^k \in \Omega, \quad & \theta(u) - \theta(u^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \\ & \geq (v - \widetilde{v}^k)^T HM(v^k - \widetilde{v}^k) \\ & \geq (v - \widetilde{v}^k)^T H(v^k - v^{k+1}), \quad \forall w \in \Omega. \end{aligned} \tag{2.3.4}$$

By **Proposition 1.1.1** in preliminaries, we have

$$\begin{aligned} & \theta(u) - \theta(u^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \\ \geq & \frac{1}{2}\|v - v^{k+1}\|_H^2 - \frac{1}{2}\|v - v^k\|_H^2 + \frac{1}{2}\|v^k - \widetilde{v}^k\|_H^2 - \frac{1}{2}\|v^{k+1} - \widetilde{v}^k\|_H^2. \end{aligned} \tag{2.3.5}$$

Since (2.2.2), $HM = Q$ and $G = Q^T + Q - M^T HM$, we have

$$\begin{aligned}
&\frac{1}{2}\|v^k - \widetilde{v}^k\|_H^2 - \frac{1}{2}\|v^{k+1} - \widetilde{v}^k\|_H^2 \\
=&\frac{1}{2}\|v^k - \widetilde{v}^k\|_H^2 - \frac{1}{2}\|(v^k - \widetilde{v}^k) - M(v^k - \widetilde{v}^k)\|_H^2 \\
=&\frac{1}{2}((v^k - \widetilde{v}^k)^T(HM + M^T H - M^T HM)(v^k - \widetilde{v}^k)) \\
=&\frac{1}{2}\|v^k - \widetilde{v}^k\|_G^2.
\end{aligned} \tag{2.3.6}$$

Substituting (2.3.6) into right hand of (2.3.5), we get (2.3.3). □

Theorem 2.3.1. *For variational inequality problem (2.1.5), assume (2.3.1) and (2.3.2) are satisfied, the sequence $\{v^k\}$ generated by prediction-correction algorithm framework has contractive property*

$$\|v^{k+1} - v^*\|_H^2 \le \|v^k - v^*\|_H^2 - \|v^k - \widetilde{v}^k\|_G^2, \quad \forall v^* \in V^*, \tag{2.3.7}$$

where V^* is optimal point set of v.

Proof. Substituting w^* for w in (2.3.3), we have

$$\begin{aligned}
&\theta(u^*) - \theta(\widetilde{u}^k) + (w^* - \widetilde{w}^k)^T F(\widetilde{w}^k) \\
\ge&\frac{1}{2}(\|v^* - v^{k+1}\|_H^2 - \|v^* - v^k\|_H^2) + \frac{1}{2}\|v^k - \widetilde{v}^k\|_G^2.
\end{aligned} \tag{2.3.8}$$

By **Remark 2.1.1**, we have

$$\begin{aligned}
&\theta(u^*) - \theta(\widetilde{u}^k) + (w^* - \widetilde{w}^k)^T F(w^*) \\
\ge&\frac{1}{2}(\|v^* - v^{k+1}\|_H^2 - \|v^* - v^k\|_H^2) + \frac{1}{2}\|v^k - \widetilde{v}^k\|_G^2.
\end{aligned} \tag{2.3.9}$$

By optimal condition (2.1.5), the theorem is proved □

Remark 2.3.1. ***Theorem 2.3.1*** *indicates that the sequence $\{v^k\}$ is contractive under H-norm.*

By (2.1.5), we know if $\widetilde{w}$ satisfies

$$\widetilde{w} \in \Omega, \quad \theta(u) - \theta(\widetilde{u}) + (w - \widetilde{w})^T F(\widetilde{w}) \ge 0, \quad \forall w \in \Omega, \tag{2.3.10}$$

$\widetilde{w}$ is a solution of variational inequality (2.1.5).

By **Remark 2.1.1**, (2.3.10) is equivalent to

$$\widetilde{w} \in \Omega, \quad \theta(u) - \theta(\widetilde{u}) + (w - \widetilde{w})^T F(w) \ge 0, \quad \forall w \in \Omega. \tag{2.3.11}$$

For each $\epsilon > 0$, if

$$\widetilde{w} \in \Omega, \quad \theta(u) - \theta(\widetilde{u}) + (w - \widetilde{w})^T F(w) \ge -\epsilon, \quad \forall w \in \mathcal{D}_{(\widetilde{w})}, \tag{2.3.12}$$

where

$$\mathcal{D}_{(\widetilde{w})} = \{w \in \Omega | \|w - \widetilde{w}\| \le 1\},$$

then $\widetilde{w} \in \Omega$ is called a ϵ-approximation solution of (2.1.5).

Next theorem shows that the prediction-correction algorithm framework also establishs a worst-case $O(\frac{1}{n})$ iteration complexity in ergodic sense, which means that we can find a $\widetilde{w} \in \Omega$ such that

$$\sup_{w \in \mathcal{D}_{(\widetilde{w})}} \{\theta(\widetilde{u}) - \theta(u) + (\widetilde{w} - w)^T F(w)\} \leq \epsilon = O(\frac{1}{n}). \tag{2.3.13}$$

Theorem 2.3.2. *For variational inequality problem (2.1.5), assume (2.3.1) and (2.3.2) are satisfied, the sequence $\{v^k\}$ generated by prediction-correction algorithm framework has worst-case $O(\frac{1}{n})$ convergence rate, that is, for each integer n, there exists*

$$\theta(\widetilde{u}_n) - \theta(u) + (\widetilde{w}_n - w)^T F(w) \leq \frac{1}{2(n+1)} \|v - v^0\|_H^2, \quad \forall w \in \Omega, \tag{2.3.14}$$

where $\widetilde{w}_n = \frac{1}{n+1}\left(\sum_{n}^{k=0} \widetilde{w}^k\right)$.

Proof. By **Remark 2.1.1** and (2.3.3), we have

$$\theta(\widetilde{u}^k) - \theta(u) + (\widetilde{w}^k - w)^T F(w) + \frac{1}{2}\|v - v^{k+1}\|_H^2 \leq \frac{1}{2}\|v - v^k\|_H^2, \quad \forall w \in \Omega. \tag{2.3.15}$$

Adding up (2.4.2) from $k = 0$ to $k = n$, we get

$$\frac{1}{n+1}\sum_{k=0}^{n} \theta(\widetilde{u}^k) - \theta(u) + (\widetilde{w}_n - w)^T F(w) \leq \frac{1}{2(n+1)} \|v - v^0\|_H^2. \tag{2.3.16}$$

Since θ is convex, we have $\theta(\frac{1}{n+1}\sum_{k=0}^{n} \widetilde{u}^k) \leq \frac{1}{n+1}\sum_{k=0}^{n} \theta(\widetilde{u}^k)$. □

Remark 2.3.2. *After n times iteration, $\widetilde{w}_n$ satisfies*

$$\widetilde{w}_n \in \Omega, \quad \sup_{w \in \mathcal{D}_{(\widetilde{w})}} \{\theta(\widetilde{u}) - \theta(u) + (\widetilde{w} - w)^T F(w)\} \leq \frac{d}{2n} = O(\frac{1}{n}), \tag{2.3.17}$$

where $d \coloneqq \sup\{\|v - v^0\|_H^2 | w \in \mathcal{D}_{(\widetilde{w})}\}$.

Theorem 2.3.2 *means that the arithmetic average of predictive point $\widetilde{w}^k$ is a ϵ-approximation point of variational inequality (2.1.5).*

Chapter 3 Improved Algorithms

As mentioned in backgrounds, the direct extension ADMM for three separable operators(1.2.5),

$$\begin{cases} x^{k+1} = \arg\min\{L_\beta(x, y^k, z^k, \lambda^k)|x \in \mathcal{X}\}, \\ y^{k+1} = \arg\min\{L_\beta(x^{k+1}, y, z^k, \lambda^k)|y \in \mathcal{Y}\}, \\ z^{k+1} = \arg\min\{L_\beta(x^{k+1}, y^{k+1}, z, \lambda^k)|z \in \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b), \end{cases} \tag{3.0.1}$$

may not be convergent, even we have some strict assumptions on objective function.

Remark 3.0.3. *It should be noticed that x^k don't participate in the iteration of y-step and z-step in (3.1.1). Intuitively, y-iteration step only uses the updated variable x^{k+1} and original variable z^k to generate y^{k+1} but z-iteration step uses both updated variable x^{k+1} and y^{k+1} to generate z^{k+1}, which would cause an imbalance in y, z-iteration.*

It is a natural thought to improve the direct extension method such that convergence could be proved under some mild assumptions. In this chapter, we will give two improved algorithms and a variant of the second method. Since we have done convergence analysis of prediction-correction framework in Chapter 2, we only need to verify whether these algorithms satisfy the assumptions of framework.

3.1 Corrected Method

The corrected method uses direct extension ADMM to iterate a predictive point and adds an extra correction step to balance y, z-iteration. Intuitively, this thought is natural and reasonable. In this method, we correct y, z-step by

$$\begin{pmatrix} y^{k+1} \\ z^{k+1} \end{pmatrix} := \begin{pmatrix} y^k \\ z^k \end{pmatrix} - \nu \begin{pmatrix} I & -(B^TB)^{-1}B^TC \\ 0 & I \end{pmatrix} \begin{pmatrix} y^k - y^{k+1} \\ z^k - z^{k+1} \end{pmatrix}, \quad \nu \in (0, 1), \tag{3.1.1}$$

where $(:=)$ means (y^{k+1}, z^{k+1}) in the right hand is generated by the direct extension of ADMM.

We rewrite (3.0.1) into an equivalent form

$$\begin{cases} x^{k+1} = \arg\min\{\theta_1(x) - (\lambda^k)^T(Ax) + \frac{\beta}{2}\|Ax + By^k + Cz^k - b\|^2|x \in \mathcal{X}\}, \\ y^{k+1} = \arg\min\{\theta_2(y) - (\lambda^k)^T(By) + \frac{\beta}{2}\|Ax^{k+1} + By + Cz^k - b\|^2|y \in \mathcal{Y}\}, \\ z^{k+1} = \arg\min\{\theta_3(z) - (\lambda^k)^T(Cz) + \frac{\beta}{2}\|Ax^{k+1} + By^{k+1} + Cz - b\|^2|z \in \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b). \end{cases} \tag{3.1.2}$$

We use $(x^{k+1}, y^{k+1}, z^{k+1})$ generated by (3.1.2) to define predictive variable $\widetilde{w}^k = (\widetilde{x}^k, \widetilde{y}^k, \widetilde{z}^k, \widetilde{\lambda}^k)$, where

$$(\widetilde{x}^k, \widetilde{y}^k, \widetilde{z}^k) = (x^{k+1}, y^{k+1}, z^{k+1}), \tag{3.1.3}$$

and

$$\widetilde{\lambda}^k = \lambda^k - \beta(A\widetilde{x}^k + By^k + Cz^k - b). \tag{3.1.4}$$

Lemma 3.1.1. *Let* $(x^{k+1}, y^{k+1}, z^{k+1})$ *be generated by (3.1.2), then the predictive point* $\widetilde{w}^k$ *satisfies*

$$\widetilde{w}^k \in \Omega, \quad \theta(u) - \theta(u^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \geq (v - \widetilde{v}^k)^T Q(v^k - \widetilde{v}^k), \quad \forall w \in \Omega, \tag{3.1.5}$$

where

$$Q = \begin{pmatrix} \beta B^T B & 0 & 0 \\ \beta C^T B & \beta C^T C & 0 \\ -B & -C & \frac{1}{\beta} I_m \end{pmatrix}. \tag{3.1.6}$$

Proof. Using (3.1.3) and (3.1.4), we can rewrite x-subproblem as

$$\widetilde{x}^k \in \mathcal{X}, \quad \theta_1(x) - \theta_1(\widetilde{x}^k) + (x - \widetilde{x}^k)^T(-A^T\widetilde{\lambda}^k) \geq 0, \quad \forall x \in \mathcal{X}. \tag{3.1.7}$$

Similarly, y-subproblem and z-subproblem can be rewrote as

$$\begin{aligned} \widetilde{y}^k \in \mathcal{Y}, \quad & \theta_2(y) - \theta_2(\widetilde{y}^k) + (y - \widetilde{y}^k)^T(-B^T\widetilde{\lambda}^k) \\ & \geq (y - \widetilde{y}^k)^T \beta B^T B(y^k - \widetilde{y}^k), \quad \forall y \in \mathcal{Y}. \end{aligned} \tag{3.1.8}$$

$$\begin{aligned} \widetilde{z}^k \in \mathcal{Z}, \quad & \theta_3(z) - \theta_3(\widetilde{z}^k) + (z - \widetilde{z}^k)^T(-C^T\widetilde{\lambda}^k) \\ & \geq (z - \widetilde{z}^k)^T(\beta C^T B(y^k - \widetilde{y}^k) + \beta C^T C(z^k - \widetilde{z}^k)), \quad \forall z \in \mathcal{Z}. \end{aligned} \tag{3.1.9}$$

By (3.1.4), we also have identity

$$(A\widetilde{x}^k + B\widetilde{y}^k + C\widetilde{z}^k - b) - B(\widetilde{y}^k - y^k) - C(\widetilde{z}^k - z^k) + \frac{1}{\beta}(\widetilde{\lambda}^k - \lambda^k) = 0. \tag{3.1.10}$$

Rewriting (3.1.10) in a form of variational inequality, we have

$$\begin{aligned} \widetilde{\lambda}^k \in \Re^m, \quad & (\lambda - \widetilde{\lambda}^k)^T(A\widetilde{x}^k + B\widetilde{y}^k + C\widetilde{z}^k - b) \\ & \geq (\lambda - \widetilde{\lambda}^k)^T\{-B(\widetilde{y}^k - y^k) - C(\widetilde{z}^k - z^k) + \frac{1}{\beta}(\widetilde{\lambda}^k - \lambda^k)\}, \quad \forall \lambda \in \Re^m. \end{aligned} \tag{3.1.11}$$

Reformulating (3.1.7), (3.1.8), (3.1.9) and (3.1.11) into a compact form and the lemma is proved. □

By (3.1.10), we have

$$\begin{aligned} \lambda^{k+1} &= \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b) \\ &= \lambda^k - [-\beta B(\widetilde{y}^k - y^k) - \beta C(\widetilde{z}^k - z^k) + (\widetilde{\lambda}^k - \lambda^k)]. \end{aligned} \tag{3.1.12}$$

Combining with (3.1.1), we get

$$v^{k+1} = v^k - M(v^k - \widetilde{v}^k), \tag{3.1.13}$$

where

$$M = \begin{pmatrix} \nu I & -\nu(B^T B)^{-1} B^T C & 0 \\ 0 & \nu I & 0 \\ -\beta B & -\beta C & I_m \end{pmatrix}. \tag{3.1.14}$$

We have reformulated corrected method as a form in prediction-correction framework, then the next task is to verify whether assumptions is established.

Firstly, rewriting matrix Q into a partitioned form

$$Q = \begin{pmatrix} \beta Q_0 & 0 \\ -\mathcal{A} & \frac{1}{\beta} I_m \end{pmatrix},$$

where

$$Q_0 = \begin{pmatrix} B^T B & 0 \\ C^T B & C^T C \end{pmatrix}, \quad \mathcal{A} = (B, C).$$

Let

$$D_0 = \begin{pmatrix} B^T B & 0 \\ 0 & C^T C \end{pmatrix} \succ 0,$$

we have $Q_0^T + Q_0 = D_0 + \mathcal{A}^T \mathcal{A}$.

Using these notations, matrix M in (3.1.14) can be wrote as

$$M = \begin{pmatrix} \nu Q_0^{-T} D_0 & 0 \\ -\beta \mathcal{A} & I_m \end{pmatrix}, \text{the inverse matrix is} \quad M^{-1} = \begin{pmatrix} \frac{1}{\nu} D_0^{-1} Q_0^T & 0 \\ \frac{1}{\nu} \beta \mathcal{A} D_0^{-1} Q_0^T & I_m \end{pmatrix}.$$

Since

$$\begin{aligned} H = QM^{-1} &= \begin{pmatrix} \beta Q_0 & 0 \\ -\mathcal{A} & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} \frac{1}{\nu} D_0^{-1} Q_0^T & 0 \\ \frac{1}{\nu} \beta \mathcal{A} D_0^{-1} Q_0^T & I_m \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{\nu} \beta Q_0 D_0^{-1} Q_0^T & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \succ 0, \end{aligned} \tag{3.1.15}$$

the assumption (2.3.1) is satisfied.

By $HM = Q$,

$$\begin{aligned} G =& Q^T + Q - M^T H M \\ =& Q^T + Q - Q^T M \\ =& \begin{pmatrix} \beta(Q_0^T + Q_0) & -\mathcal{A}^T \\ -\mathcal{A} & \frac{2}{\beta} I_m \end{pmatrix} - \begin{pmatrix} \beta Q_0^T & -\mathcal{A}^T \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} \nu Q_0^{-T} D_0 & 0 \\ -\beta \mathcal{A} & I_m \end{pmatrix} \\ =& \begin{pmatrix} \beta(Q_0^T + Q_0) & -\mathcal{A}^T \\ -\mathcal{A} & \frac{2}{\beta} I_m \end{pmatrix} - \begin{pmatrix} \beta(\nu D_0 + \mathcal{A}^T \mathcal{A}) & -\mathcal{A}^T \\ -\mathcal{A} & \frac{1}{\beta} I_m \end{pmatrix} \\ =& \begin{pmatrix} \beta(1-\nu) D_0 & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix}, \end{aligned} \tag{3.1.16}$$

then for $\nu \in (0,1)$, G is positive definite and satisfies the assumption (2.3.2).

We can summarize the above verification as the following theorem.

Theorem 3.1.1. *For $w^k = (x^k, y^k, z^k, \lambda^k)$ generated by corrected method, assume $\nu \in (0,1)$, v^k has contractive property*

$$\|v^{k+1} - v^*\|_H^2 \leq \|v^k - v^*\|_H^2 - \|v^k - \widetilde{v}^k\|_G^2, \quad \forall v^* \in V^*, \tag{3.1.17}$$

where V^* is optimal point set of v, and has worst-case $O(\frac{1}{n})$ convergence rate, that is, for each integer n, there exists

$$\theta(\widetilde{u}_n) - \theta(u) + (\widetilde{w}_n - w)^T F(w) \le \frac{1}{2(n+1)} \|v - v^0\|_H^2, \quad \forall w \in \Omega, \tag{3.1.18}$$

where $\widetilde{w}_n = \frac{1}{n+1}\left(\sum_n^{k=0} \widetilde{w}^k\right)$.

Remark 3.1.1. *We can let $\nu = 1$ in correction step. At this time, matrix G is positive semi-definite and **Theorem 3.1.1** is also established, but we cannot guarantee that w^k will be closer to optimal point at each iteration.*

3.2 Equalized Method

The equalized method is another idea to improve the direct extension of ADMM. Since the imbalance in each iteration of direct extension method is caused by y, z-step, it is a natural work to use y^k, not y^{k+1}, to participate in z-iteration. This would forcefully equalize update of y-iteration and z-iteration, because the information used for updating is identical at each step.

This equalized practice is similar to relaxing simultaneous iteration of (y, z), that is,

$$\begin{cases} x^{k+1} = \arg\min\{L_\beta(x, y^k, z^k, \lambda^k) | x \in \mathcal{X}\}, \\ (y^{k+1}, z^{k+1}) = \arg\min\{L_\beta(x^{k+1}, y, z, \lambda^k) | (y, z) \in \mathcal{Y} \times \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b). \end{cases} \tag{3.2.1}$$

(3.2.1) is convergent because it is ADMM for $(x, (y, z))$. Equalized method is to relax (y, z)-step and iterate y and z separately with an quadratic penalty term, that is,

$$\begin{cases} x^{k+1} = \arg\min\{L_\beta(x, y^k, z^k, \lambda^k) | x \in \mathcal{X}\}, \\ y^{k+1} = \arg\min\{L_\beta(x^{k+1}, y, z^k, \lambda^k) + \frac{\tau\beta}{2}\|B(y - y^k)\|^2 | y \in \mathcal{Y}\}, \\ z^{k+1} = \arg\min\{L_\beta(x^{k+1}, y^k, z, \lambda^k) + \frac{\tau\beta}{2}\|C(z - z^k)\|^2 | z \in \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b), \end{cases} \tag{3.2.2}$$

where $\tau > 1$.

We can rewrite (3.2.2) into an equivalent form

$$\begin{cases} x^{k+1} = \arg\min\{\theta_1(x) - (\lambda^k)^T(Ax) + \frac{\beta}{2}\|Ax + By^k + Cz^k - b\|^2 | x \in \mathcal{X}\}, \\ \lambda^{k+\frac{1}{2}} = \lambda^k - \beta(Ax^{k+1} + By^k + Cz^k - b), \\ y^{k+1} = \arg\min\{\theta_2(y) - (\lambda^{k+\frac{1}{2}})^T(By) + \frac{\mu\beta}{2}\|B(y - y^k)\|^2 | y \in \mathcal{Y}\}, \\ z^{k+1} = \arg\min\{\theta_3(z) - (\lambda^{k+\frac{1}{2}})^T(Cz) + \frac{\mu\beta}{2}\|C(z - z^k)\|^2 | z \in \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b), \end{cases} \tag{3.2.3}$$

where $\mu > 2$

Same as corrected method, we only verify that algorithm (3.2.3) satisfies assumptions of prediction-correction framework.

Firstly, we define predictive variable $\widetilde{w}^k = (\widetilde{x}^k, \widetilde{y}^k, \widetilde{z}^k, \widetilde{\lambda}^k)$, where

$$(\widetilde{x}^k, \widetilde{y}^k, \widetilde{z}^k) = (x^{k+1}, y^{k+1}, z^{k+1}), \tag{3.2.4}$$

and

$$\widetilde{\lambda}^k = \lambda^k - \beta(A\widetilde{x}^k + By^k + Cz^k - b). \tag{3.2.5}$$

Lemma 3.2.1. *Let* $(x^{k+1}, y^{k+1}, z^(k+1))$ *be generated by algorithm (3.2.3), then* $\widetilde{w}^k$ *satisfies*

$$\widetilde{w}^k \in \Omega, \quad \theta(u) - \theta(u^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \geq (v - \widetilde{v}^k)^T Q(v^k - \widetilde{v}^k), \quad \forall w \in \Omega, \tag{3.2.6}$$

where

$$Q = \begin{pmatrix} \mu\beta B^T B & 0 & 0 \\ 0 & \mu\beta C^T C & 0 \\ -B & -C & \frac{1}{\beta} I_m \end{pmatrix}. \tag{3.2.7}$$

Proof. By (3.2.4) and (3.2.5), we can rewrite x-subproblem as

$$\widetilde{x}^k \in \mathcal{X}, \quad \theta_1(x) - \theta_1(\widetilde{x}^k) + (x - \widetilde{x}^k)^T(-A^T\widetilde{\lambda}^k) \geq 0, \quad \forall x \in \mathcal{X}. \tag{3.2.8}$$

Since $\widetilde{\lambda}^k = \lambda^{k+\frac{1}{2}}$, y, z-subproblem can be rewritten as

$$\begin{aligned} \widetilde{y}^k \in \mathcal{Y}, \quad & \theta_2(y) - \theta_2(\widetilde{y}^k) + (y - \widetilde{y}^k)^T(-B^T\widetilde{\lambda}^k) \\ & \geq (y - \widetilde{y}^k)^T \mu\beta B^T B(y^k - \widetilde{y}^k), \quad \forall y \in \mathcal{Y}, \end{aligned} \tag{3.2.9}$$

and

$$\begin{aligned} \widetilde{z}^k \in \mathcal{Z}, \quad & \theta_3(z) - \theta_3(\widetilde{z}^k) + (z - \widetilde{z}^k)^T(-C^T\widetilde{\lambda}^k) \\ & \geq (z - \widetilde{z}^k)^T \mu\beta C^T C(z^k - \widetilde{z}^k), \quad \forall z \in \mathcal{Z}. \end{aligned} \tag{3.2.10}$$

Also, by identity

$$(A\widetilde{x}^k + B\widetilde{y}^k + C\widetilde{z}^k - b) - B(\widetilde{y}^k - y^k) - C(\widetilde{z}^k - z^k) + \frac{1}{\beta}(\widetilde{\lambda}^k - \lambda^k) = 0, \tag{3.2.11}$$

we can write it in a form of variational inequality

$$\begin{aligned} \widetilde{\lambda}^k \in \Re^m, \quad & (\lambda - \widetilde{\lambda}^k)^T(A\widetilde{x}^k + B\widetilde{y}^k + C\widetilde{z}^k - b) \\ & \geq (\lambda - \widetilde{\lambda}^k)^T\{-B(\widetilde{y}^k - y^k) - C(\widetilde{z}^k - z^k) + \frac{1}{\beta}(\widetilde{\lambda}^k - \lambda^k)\}, \quad \forall \lambda \in \Re^m. \end{aligned} \tag{3.2.12}$$

Combining (3.2.8), (3.2.9), (3.2.10) and (3.2.12), we have completed the prove. □

Defining corrective variable v^{k+1} as

$$y^{k+1} = \widetilde{y}^k, \quad z^{k+1} = \widetilde{z}^k, \tag{3.2.13}$$

and

$$\begin{aligned} \lambda^{k+1} =& \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b) \\ =& \lambda^k - [-\beta B(y^k - \widetilde{y}^k) - \beta C(z^k - \widetilde{z}^k) + (\lambda^k - \widetilde{\lambda}^k)], \end{aligned} \tag{3.2.14}$$

we have

$$\begin{pmatrix} y^{k+1} \\ z^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^{k} \\ z^{k} \\ \lambda^{k} \end{pmatrix} - \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\beta B & -\beta C & I_m \end{pmatrix} \begin{pmatrix} y^k - \widetilde{y}^k \\ z^k - \widetilde{z}^k \\ \lambda^k - \widetilde{\lambda}^k \end{pmatrix}. \tag{3.2.15}$$

In other words, we have

$$v^{k+1} = v^k - M(v^k - \widetilde{v}^k), \quad \text{where} \quad M = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -\beta B & -\beta C & I_m \end{pmatrix}. \tag{3.2.16}$$

Since

$$\begin{aligned} H = QM^{-1} &= \begin{pmatrix} \mu\beta B^T B & 0 & 0 \\ 0 & \mu\beta C^T C & 0 \\ -B & -C & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \beta B & \beta C & I_m \end{pmatrix} \\ &= \begin{pmatrix} \mu\beta B^T B & 0 & 0 \\ 0 & \mu\beta C^T C & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix}, \end{aligned} \tag{3.2.17}$$

H is positive definite with assumptions that B and C have full column rank.

By $Q = HM$ and $G = Q^T + Q - Q^T M$,

$$\begin{aligned} G &= \begin{pmatrix} 2\mu\beta B^T B & 0 & -B^T \\ 0 & 2\mu\beta C^T C & -C^T \\ -B & -C & \frac{2}{\beta} I_m \end{pmatrix} - \begin{pmatrix} (\mu+1)\beta B^T B & -\beta B^T C & -B^T \\ -\beta C^T B & (\mu+1)\beta C^T C & -C^T \\ -B & -C & \frac{1}{\beta} I_m \end{pmatrix} \\ &= \begin{pmatrix} (\mu-1)\beta B^T B & \beta B^T C & 0 \\ \beta C^T B & (\mu-1)\beta C^T C & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix} \\ &= \begin{pmatrix} (\mu-2)\beta B^T B & 0 & 0 \\ 0 & (\mu-2)\beta C^T C & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix} + \begin{pmatrix} \beta B^T B & \beta B^T C & 0 \\ \beta C^T B & \beta C^T C & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} (\mu-2)\beta B^T B & 0 & 0 \\ 0 & (\mu-2)\beta C^T C & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix} + \beta \begin{pmatrix} B^T \\ C^T \\ 0 \end{pmatrix} \begin{pmatrix} B & C & 0 \end{pmatrix}, \end{aligned} \tag{3.2.18}$$

G is positive definite for $\mu > 2$.

We have put equalized method into prediction-correction framework, and the following theorem is established.

Theorem 3.2.1. *For $w^k = (x^k, y^k, z^k, \lambda^k)$ generated by equalized method (3.2.3), assume $\mu > 2$, v^k has contractive property*

$$\|v^{k+1} - v^*\|_H^2 \leq \|v^k - v^*\|_H^2 - \|v^k - \widetilde{v}^k\|_G^2, \quad \forall v^* \in V^*, \tag{3.2.19}$$

where V^* is optimal point set of v, and has worst-case $O(\frac{1}{n})$ convergence rate, that is, for each integer n, there exists

$$\theta(\widetilde{u}_n) - \theta(u) + (\widetilde{w}_n - w)^T F(w) \leq \frac{1}{2(n+1)}\|v - v^0\|_H^2, \quad \forall w \in \Omega, \tag{3.2.20}$$

where $\widetilde{w}_n = \frac{1}{n+1}\left(\sum_n^{k=0} \widetilde{w}^k\right)$.

3.3 Variant of Equalized Method

In this section, we will give a variant of equalized method, but detailed proof will be omitted because it is not a natural improved algorithm for direction extension ADMM method.

According to the proof of equalized method, it is not difficult to see that if we forcefully equalize x, y-subproblems, similar structure of variational inequality will be reformulated, and we can also put it into prediction-correction framework to consider.

However, we should make a little revision on framework (2.2.1) and (2.2.2) respectively, that is,

Prediction: we use a provided w^k to generate a $\widetilde{w}^k$ such that

$$\widetilde{w}^k \in \Omega, \quad \theta(u) - \theta(u^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \geq (w - \widetilde{w}^k)^T Q(w^k - \widetilde{w}^k), \quad \forall w \in \Omega. \tag{3.3.1}$$

Correction: the new iterative point w^{k+1} is generated by

$$w^{k+1} = w^k - M(w^k - \widetilde{w}^k). \tag{3.3.2}$$

It is easy to verify that this revised framework also has similar contractive property and $O(\frac{1}{n})$ convergence rate in ergodic sense for w^k under same assumptions, that is, assume $Q^T + Q$ in (3.3.1) is positive definite, and for matrix M in (3.3.2), there exists a positive definite matrix H such that

$$HM = Q, \tag{3.3.3}$$

and

$$G = Q^T + Q - M^T HM \succ 0, \quad \text{(at least positive semi-definite)}. \tag{3.3.4}$$

The following two theorems are also established.

Theorem 3.3.1. *Assume (3.3.3) and (3.3.4) are satisfied, the sequence $\{w^k\}$ generated by (3.3.1) and (3.3.2) has contractive property*

$$\|w^{k+1} - w^*\|_H^2 \leq \|w^k - w^*\|_H^2 - \|w^k - \widetilde{w}^k\|_G^2, \quad \forall w^* \in W^*, \tag{3.3.5}$$

where W^* is optimal point set of w.

Theorem 3.3.2. *Assume (3.3.3) and (3.3.4) are satisfied, the sequence $\{w^k\}$ generated by (3.3.1) and (3.3.2) has worst-case $O(\frac{1}{n})$ convergence rate.*

The variant algorithm is stated as

$$\begin{cases} x^{k+1} = \arg\min\{L_\beta(x, y^k, z^k, \lambda^k) + \frac{\tau\beta}{2}\|A(x-x^k)\|^2 | x \in \mathcal{X}\}, \\ y^{k+1} = \arg\min\{L_\beta(x^k, y, z^k, \lambda^k) + \frac{\tau\beta}{2}\|B(y-y^k)\|^2 | y \in \mathcal{Y}\}, \\ z^{k+1} = \arg\min\{L_\beta(x^{k+1}, y^{k+1}, z, \lambda^k) | z \in \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b), \end{cases} \tag{3.3.6}$$

where $\tau > 1$.

Define predictive variable $\widetilde{w}^k = (\widetilde{x}^k, \widetilde{y}^k, \widetilde{z}^k, \widetilde{\lambda}^k)$ as

$$(\widetilde{x}^k, \widetilde{y}^k, \widetilde{z}^k) = (x^{k+1}, y^{k+1}, z^{k+1}), \tag{3.3.7}$$

and

$$\widetilde{\lambda}^k = \lambda^k - \beta(A\widetilde{x}^k + B\widetilde{y}^k + Cz^k - b). \tag{3.3.8}$$

(3.3.6) is equivalent to

$$\begin{cases} \theta_1(x) - \theta_1(\widetilde{x}^k) + (x - \widetilde{x}^k)^T[-A^T\widetilde{\lambda}^k - \beta A^T B(\widetilde{y}^k - y^k) + \tau\beta A^T A(\widetilde{x}^k - x^k)] \geq 0, \quad \forall x \in \mathcal{X}, \\ \theta_2(y) - \theta_2(\widetilde{y}^k) + (y - \widetilde{y}^k)^T[-B^T\widetilde{\lambda}^k - \beta B^T A(\widetilde{x}^k - x^k) + \tau\beta B^T B(\widetilde{y}^k - y^k)] \geq 0, \quad \forall y \in \mathcal{Y}, \\ \theta_3(z) - \theta_3(\widetilde{z}^k) + (z - \widetilde{z}^k)^T[-C^T\widetilde{\lambda}^k + \beta C^T C(\widetilde{z}^k - z^k)] \geq 0, \quad \forall z \in \mathcal{Z}, \\ (\lambda - \widetilde{\lambda}^k)^T[(A\widetilde{x}^k + B\widetilde{y}^k + C\widetilde{z}^k - b) - C(\widetilde{z}^k - z^k) + \frac{1}{\beta}(\widetilde{\lambda}^k - \lambda^k)] \geq 0, \quad \forall \lambda \in \Re^m, \end{cases} \tag{3.3.9}$$

where $\tau > 1$.

Rewriting (3.3.9) into a compact form, we obtain

$$\begin{aligned} \widetilde{w}^k \in \Omega, \quad & \theta(u) - \theta(u^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \\ \geq & (w - \widetilde{w}^k)^T \begin{pmatrix} \tau\beta A^T A & -\beta A^T B & 0 & 0 \\ -\beta B^T A & \tau\beta B^T B & 0 & 0 \\ 0 & 0 & \beta C^T C & 0 \\ 0 & 0 & -C & \frac{1}{\beta} I_m \end{pmatrix} (w^k - \widetilde{w}^k), \quad \forall w \in \Omega, \end{aligned} \tag{3.3.10}$$

where $\tau > 1$.

By (3.3.7) and (3.3.8), defining corrective variable w^{k+1} as

$$x^{k+1} = \widetilde{x}^k, \quad y^{k+1} = \widetilde{y}^k, \quad z^{k+1} = \widetilde{z}^k, \tag{3.3.11}$$

and

$$\lambda^{k+1} = \lambda^k - [-\beta C(z^k - \widetilde{z}^k) + (\lambda^k - \widetilde{\lambda}^k)], \tag{3.3.12}$$

we have

$$w^{k+1} = w^k - \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & -\beta C & I_m \end{pmatrix} (w^k - \widetilde{w}^k). \tag{3.3.13}$$

Since $H = QM^{-1}$ and

$$M^{-1} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & \beta C & I_m \end{pmatrix}, \tag{3.3.14}$$

then

$$H = \begin{pmatrix} \tau\beta A^T A & -\beta A^T B & 0 & 0 \\ -\beta B^T A & \tau\beta B^T B & 0 & 0 \\ 0 & 0 & \beta C^T C & 0 \\ 0 & 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix} \succ 0 \tag{3.3.15}$$

with $\tau > 1$, and

$$G = Q^T + Q - M^T H M = \begin{pmatrix} \tau\beta A^T A & -\beta A^T B & 0 & 0 \\ -\beta B^T A & \tau\beta B^T B & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix} \succeq 0. \tag{3.3.16}$$

Remark 3.3.1. *It should be noticed that, in ADMM and direct extension ADMM method, x^k do not participate in the iteration of y, z and λ-step. Corrected method and equalized method also inherit this manner. Therefore, these two improved algorithms are more ADMM-like. In this section, algorithm (3.3.6) uses x^k to iterate y^{k+1}, thus, it is kind of different from previous two improved algorithms.*

Chapter 4 Accelerated Algorithms

In Chapter 3, we give two methods to improve the direct extension of ADMM for three separable operators. A variant of equalized method is also mentioned because it can be considered in similar prediction-correction algorithm framework. In this chapter, we will focus on equalized method and give an accelerated algorithm of it.

4.1 Equalized Method with Relaxation Factor

For original equalized method, the problem is stated as

$$\begin{cases} x^{k+1} = \arg\min\{L_\beta(x, y^k, z^k, \lambda^k) | x \in \mathcal{X}\}, \\ y^{k+1} = \arg\min\{L_\beta(x^{k+1}, y, z^k, \lambda^k) + \frac{\tau\beta}{2}\|B(y - y^k)\|^2 | y \in \mathcal{Y}\}, \\ z^{k+1} = \arg\min\{L_\beta(x^{k+1}, y^k, z, \lambda^k) + \frac{\tau\beta}{2}\|C(z - z^k)\|^2 | z \in \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b), \end{cases} \tag{4.1.1}$$

where $\tau > 1$.

As introduced in backgrounds of chapter 1, we can add a relaxation factor γ on λ-step in ADMM such that

$$\lambda^{k+1} = \lambda^k - \gamma\beta(Ax^{k+1} + By^{k+1} - b). \tag{4.1.2}$$

It is not difficult to use variational inequality to prove that ADMM with relaxation factor is convergent with $\gamma \in (0, \frac{1+\sqrt{5}}{2})$. Since the relaxation factor γ can generate the step size larger than 1, it usually accelerates convergence of ADMM. The detailed proof will be included in Appendix A.

Since equalize method is ADMM-like, it is natural to think whether it is also convergent to treat equalized method in the same manner, that is,

$$\begin{cases} x^{k+1} = \arg\min\{L_\beta(x, y^k, z^k, \lambda^k) | x \in \mathcal{X}\}, \\ y^{k+1} = \arg\min\{L_\beta(x^{k+1}, y, z^k, \lambda^k) + \frac{\tau\beta}{2}\|B(y - y^k)\|^2 | y \in \mathcal{Y}\}, \\ z^{k+1} = \arg\min\{L_\beta(x^{k+1}, y^k, z, \lambda^k) + \frac{\tau\beta}{2}\|C(z - z^k)\|^2 | z \in \mathcal{Z}\}, \\ \lambda^{k+1} = \lambda^k - \gamma\beta(Ax^{k+1} + By^{k+1} + Cz^{k+1} - b), \end{cases} \tag{4.1.3}$$

where $\tau > 1$ and $\gamma \in (0, \frac{1+\sqrt{5}}{2})$.

In the following discussion, we use η to denote (y, z), $\bar{\theta}(\eta)$ to denote $\theta_2(y) + \theta_3(z)$ and $\mathfrak{B}$ to denote $[B, C]$.

Same as section 3.2, we define predictive variable as

$$\widetilde{x}^k = x^{k+1}, \quad \widetilde{y}^k = y^{k+1}, \quad \widetilde{z}^k = z^{k+1}, \tag{4.1.4}$$

and

$$\widetilde{\lambda}^k = \lambda^k - \beta(Ax^{k+1} + \mathfrak{B}\eta^k - b). \tag{4.1.5}$$

Theorem 4.1.1. *Let $w^k = (x^k, \eta^k, \lambda^k)$ be generated by (4.1.3), then predictive variable $\widetilde{w}^k$ satisfies*

$$\widetilde{w}^k \in \Omega, \quad \theta(u) - \theta(u^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \geq (v - \widetilde{v}^k)^T Q(v^k - \widetilde{v}^k), \quad \forall w \in \Omega, \tag{4.1.6}$$

where

$$Q = \begin{pmatrix} (\tau+1)\beta\mathfrak{D} & 0 \\ -\mathfrak{B} & \frac{1}{\beta} I_m \end{pmatrix}, \tag{4.1.7}$$

and

$$\mathfrak{D} = \begin{pmatrix} B^T B & 0 \\ 0 & C^T C \end{pmatrix}. \tag{4.1.8}$$

Proof. Equivalent form of (4.1.3) is

$$\begin{cases} \theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^T[-A^T\lambda^k + \beta A^T(Ax^{k+1} + \bar{B}v^k - b)] \geq 0, \quad \forall x \in \mathcal{X}, \\ \theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^T[-B^T\lambda^k + \beta B^T(Ax^{k+1} + \bar{B}v^{k+1} - b) - \beta B^T\bar{B}(v^{k+1} - v^k) \\ \quad + (\tau+1)\beta B^T B(y^{k+1} - y^k)] \geq 0, \quad \forall y \in \mathcal{Y}, \\ \theta_3(z) - \theta_3(z^{k+1}) + (z - z^{k+1})^T[-C^T\lambda^k + \beta C^T(Ax^{k+1} + \bar{B}v^{k+1} - b) - \beta C^T\bar{B}(v^{k+1} - v^k) \\ \quad + (\tau+1)\beta C^T C(z^{k+1} - z^k)] \geq 0, \quad \forall z \in \mathcal{Z}, \\ (Ax^{k+1} + By^k + Cz^k - b) + \dfrac{1}{\beta}(\widetilde{\lambda}^k - \lambda^k) = 0. \end{cases} \tag{4.1.9}$$

Combining y, z-step in (4.1.9), we have

$$\begin{cases} \theta_1(x) - \theta_1(\widetilde{x}^k) + (x - \widetilde{x}^k)^T(-A^T\widetilde{\lambda}^k) \geq 0, \quad \forall x \in \mathcal{X}, \\ \bar{\theta}(\eta) - \bar{\theta}(\widetilde{\eta}^k) + (\eta - \widetilde{\eta}^k)^T[-\mathfrak{B}^T\widetilde{\lambda}^k + (\tau+1)\beta \begin{bmatrix} B^T B & 0 \\ 0 & C^T C \end{bmatrix} (\widetilde{\eta}^k - \eta^k)] \geq 0, \quad \forall \eta \in \mathcal{Y} \times \mathcal{Z}, \\ (\lambda - \widetilde{\lambda}^k)^T[A\widetilde{x}^k + \mathfrak{B}\widetilde{\eta}^k - b - \mathfrak{B}(\widetilde{\eta}^k - \eta^k) + \dfrac{1}{\beta}(\widetilde{\lambda}^k - \lambda^k)] \geq 0, \quad \forall \lambda \in \Re^m. \end{cases} \tag{4.1.10}$$

Rewriting (4.1.10) into a compact form, we obtain (4.1.6) and the lemma is proved. □

By (4.1.3) and (4.1.4), defining corrective variable v^{k+1} as

$$\eta^{k+1} = \widetilde{\eta}^k, \tag{4.1.11}$$

and

$$\lambda^{k+1} = \lambda^k - \gamma[-\beta\mathfrak{B}(\eta^k - \widetilde{\eta}^k) + (\lambda^k - \widetilde{\lambda}^k)], \tag{4.1.12}$$

we obtain

$$\begin{pmatrix} \eta^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} \eta^k \\ \lambda^k \end{pmatrix} - M \begin{pmatrix} \eta^k - \widetilde{\eta}^k \\ \lambda^k - \widetilde{\lambda}^k \end{pmatrix}, \tag{4.1.13}$$

where

$$M = \begin{pmatrix} I & 0 \\ -\gamma\beta\mathfrak{B} & \gamma I_m \end{pmatrix}. \tag{4.1.14}$$

It is not difficult to observe that there exists a positive definite matrix H such that $Q = HM$, since

$$\begin{aligned} H = QM^{-1} &= \begin{pmatrix} (\tau+1)\beta\mathfrak{D} & 0 \\ -\mathfrak{B} & \frac{1}{\beta}I_m \end{pmatrix} \begin{pmatrix} I & 0 \\ \beta\mathfrak{B} & \frac{1}{\gamma}I_m \end{pmatrix} \\ &= \begin{pmatrix} (\tau+1)\beta\mathfrak{D} & 0 \\ 0 & \frac{1}{\gamma\beta}I_m \end{pmatrix} \end{aligned} \tag{4.1.15}$$

is positive definite with $\mathfrak{D} \succ 0$.

Lemma 4.1.1. *Let $w^k = (x^k, \eta^k, \lambda^k)$ be generated by (4.1.3), then predictive variable $\widetilde{w}^k$ satisfies*

$$\begin{aligned} &\widetilde{w}^k \in \Omega, \quad \theta(u) - \theta(u^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \\ &\quad \geq \frac{1}{2}\|v - v^{k+1}\|_H^2 - \frac{1}{2}\|v - v^k\|_H^2 + \frac{1}{2}\|\widetilde{v}^k - v^k\|_H^2 - \frac{1}{2}\|\widetilde{v}^k - v^{k+1}\|_H^2, \quad \forall w \in \Omega. \end{aligned} \tag{4.1.16}$$

Proof. By $Q = HM$, the right hand of (3.1.6) can be written as

$$(v - \widetilde{v}^k)H(v^k - v^{k+1}).$$

Thus, we have

$$\theta(u) - \theta(u^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \geq (v - \widetilde{v}^k)H(v^k - v^{k+1}). \tag{4.1.17}$$

By **Proposition 1.1.1** in preliminaries, we have

$$(v - \widetilde{v}^k)H(v^k - v^{k+1}) = \frac{1}{2}\|v - v^{k+1}\|_H^2 - \frac{1}{2}\|v - v^k\|_H^2 + \frac{1}{2}\|\widetilde{v}^k - v^k\|_H^2 - \frac{1}{2}\|\widetilde{v}^k - v^{k+1}\|_H^2. \tag{4.1.18}$$

Substituting it into (4.1.17), (4.1.16) follows directly. □

Lemma 4.1.2. *Let $w^k = (x^k, \eta^k, \lambda^k)$ be generated by (4.1.3), then we have*

$$\begin{aligned} (\eta^k - \eta^{k+1})^T \mathfrak{B}^T [(\gamma - 1)\beta(Ax^k + \mathfrak{B}\eta^k - b) &+ \beta(Ax^{k+1} + \mathfrak{B}\eta^{k+1} - b)] \\ &\geq \frac{1}{2}\|\eta^{k+1} - \eta^k\|_E^2 - \frac{1}{2}\|\eta^k - \eta^{k-1}\|_E^2, \end{aligned} \tag{4.1.19}$$

where

$$E = (\tau+1)\beta\mathfrak{D} - \beta\mathfrak{B}^T\mathfrak{B} \succ 0 \tag{4.1.20}$$

under assumptions $\tau > 1$ and $\mathfrak{D} \succ 0$.

Proof. Rewriting η-step in (4.1.10), we have

$$\bar{\theta}(\eta) - \bar{\theta}(\eta^{k+1}) + (\eta - \eta^{k+1})^T[-\mathfrak{B}^T\lambda^k + \beta\mathfrak{B}^T(Ax^{k+1} + \mathfrak{B}\eta^{k+1} - b) + E(\eta^{k+1} - \eta^k)] \geq 0. \tag{4.1.21}$$

Also, kth-iteration, we have

$$\bar{\theta}(\eta) - \bar{\theta}(\eta^k) + (\eta - \eta^k)^T[-\mathfrak{B}^T\lambda^{k-1} + \beta\mathfrak{B}^T(Ax^k + \mathfrak{B}\eta^k - b) + E(\eta^k - \eta^{k-1})] \geq 0. \tag{4.1.22}$$

Setting $\eta = \eta^k$ and $\eta = \eta^{k+1}$ respectively in (4.1.21) and (4.1.22), we obtain

$$\begin{cases} \bar{\theta}(\eta^k) - \bar{\theta}(\eta^{k+1}) + (\eta^k - \eta^{k+1})^T[-\mathfrak{B}^T\lambda^k + \beta\mathfrak{B}^T(Ax^{k+1} + \mathfrak{B}\eta^{k+1} - b) + E(\eta^{k+1} - \eta^k)] \geq 0, \\ \bar{\theta}(\eta^{k+1}) - \bar{\theta}(\eta^k) + (\eta^{k+1} - \eta^k)^T[-\mathfrak{B}^T\lambda^{k-1} + \beta\mathfrak{B}^T(Ax^k + \mathfrak{B}\eta^k - b) + E(\eta^k - \eta^{k-1})] \geq 0. \end{cases} \tag{4.1.23}$$

Adding them and we get

$$(\eta^k-\eta^{k+1})^T\mathfrak{B}^T[(\lambda^{k-1}-\lambda^k)+\beta(Ax^{k+1}+\mathfrak{B}\eta^{k+1}-Ax^k-\bar{B}\eta^k)+E(\eta^{k+1}-\eta^k)+E(\eta^{k-1}-\eta^k)]\geq 0, \tag{4.1.24}$$

that is,

$$\begin{aligned}(\eta^k-\eta^{k+1})^T\mathfrak{B}^T[(\gamma-1)\beta(Ax^k+\mathfrak{B}\eta^k-b)+&\beta(Ax^{k+1}+\mathfrak{B}\eta^{k+1}-b)]\\ &\geq\|\eta^{k+1}-\eta^k\|_E^2-(\eta^{k+1}-\eta^k)^TE(\eta^k-\eta^{k-1}).\end{aligned} \tag{4.1.25}$$

By **Proposition 1.1.1** in preliminaries, we have

$$(\eta^{k+1}-\eta^k)^TE(\eta^k-\eta^{k-1})=\frac{1}{2}\|\eta^{k+1}-\eta^{k-1}\|_E^2-\frac{1}{2}\|\eta^{k+1}-\eta^k\|_E^2-\frac{1}{2}\|\eta^k-\eta^{k-1}\|_E^2. \tag{4.1.26}$$

Hence, the right hand of (4.1.25) can be rewritten as

$$\begin{aligned}&\frac{1}{2}\|\eta^{k+1}-\eta^k\|_E^2-\frac{1}{2}\|\eta^k-\eta^{k-1}\|_E^2+\|\eta^{k+1}-\eta^k\|_E^2+\|\eta^k-\eta^{k-1}\|_E^2-\frac{1}{2}\|\eta^{k+1}-\eta^{k-1}\|_E^2\\ \geq&\frac{1}{2}\|\eta^{k+1}-\eta^k\|_E^2-\frac{1}{2}\|\eta^k-\eta^{k-1}\|_E^2,\end{aligned} \tag{4.1.27}$$

since $\|\eta^{k+1}-\eta^k\|_E^2+\|\eta^k-\eta^{k-1}\|_E^2-\frac{1}{2}\|\eta^{k+1}-\eta^{k-1}\|_E^2\geq\frac{1}{2}\|\eta^{k+1}-\eta^{k-1}\|_E^2\geq 0$.

Substituting (4.1.27) into (4.1.25) and lemma is established. □

By **Lemma 4.1.1**, we have

$$\begin{aligned}\widetilde{w}^k\in\Omega,\quad&\theta(u)-\theta(u^k)+(w-\widetilde{w}^k)^TF(\widetilde{w}^k)\\ &\geq\frac{1}{2}\|v-v^{k+1}\|_H^2-\frac{1}{2}\|v-v^k\|_H^2+\frac{1}{2}\|\widetilde{v}^k-v^k\|_H^2-\frac{1}{2}\|\widetilde{v}^k-v^{k+1}\|_H^2,\quad\forall w\in\Omega.\end{aligned}$$

The next lemma focus on the

$$\frac{1}{2}\|\widetilde{v}^k-v^k\|_H^2-\frac{1}{2}\|\widetilde{v}^k-v^{k+1}\|_H^2.$$

Lemma 4.1.3. *Let $w^k=(x^k,\eta^k,\lambda^k)$ be generated by (4.1.3) and $\gamma\in(0,\frac{1+\sqrt{5}}{2})$, we have*

$$\begin{aligned}&\|\widetilde{v}^k-v^k\|_H^2-\|\widetilde{v}^k-v^{k+1}\|_H^2\\ \geq&2\|\eta^{k+1}-\eta^k\|_E^2-\|\eta^{k-1}-\eta^k\|_E^2+(1+\gamma-\gamma^2)\beta\|Ax^{k+1}+\mathfrak{B}\eta^{k+1}-b\|^2+\\ &(1-\gamma)^2\beta[\|Ax^{k+1}+\mathfrak{B}\eta^{k+1}-b\|^2-\|Ax^k+\mathfrak{B}\eta^k-b\|^2].\end{aligned} \tag{4.1.28}$$

Proof. Since (4.1.13), we have

$$\begin{aligned}\|v^{k+1}-\widetilde{v}^k\|_H^2&=\|v^k-\widetilde{v}^k-M(v^k-\widetilde{v}^k)\|_H^2\\ &=\frac{1}{\gamma\beta}\|(\lambda^k-\widetilde{\lambda}^k)-[-\gamma\beta\mathfrak{B}(\eta^k-\widetilde{\eta}^k)+\gamma(\lambda^k-\widetilde{\lambda}^k)]\|^2\\ &=\frac{1}{\gamma\beta}\|(\lambda^k-\widetilde{\lambda}^k)-\gamma\beta(A\widetilde{x}^k+\mathfrak{B}\widetilde{\eta}^k-b)\|^2.\end{aligned} \tag{4.1.29}$$

Thus,

$$\begin{aligned}&\|v^k-\widetilde{v}^k\|_H^2-\|\widetilde{v}^k-v^{k+1}\|_H^2\\ =&\|\eta^k-\widetilde{\eta}^k\|_E^2+\beta\|\mathfrak{B}(\eta^k-\widetilde{\eta}^k)\|^2+\frac{1}{\gamma\beta}\|\lambda^k-\widetilde{\lambda}^k\|^2-\frac{1}{\gamma\beta}\|(\lambda^k-\widetilde{\lambda}^k)-\gamma\beta(A\widetilde{x}^k+\mathfrak{B}\widetilde{\eta}^k-b)\|^2\\ =&\|\eta^k-\widetilde{\eta}^k\|_E^2+\beta\|\mathfrak{B}(\eta^k-\widetilde{\eta}^k)\|^2+2(\lambda^k-\widetilde{\lambda}^k)^T(A\widetilde{x}^k+\mathfrak{B}\widetilde{\eta}^k-b)-\gamma\beta\|A\widetilde{x}^k+\mathfrak{B}\widetilde{\eta}^k-b\|^2.\end{aligned} \tag{4.1.30}$$

Since $\lambda^k - \widetilde{\lambda}^k = \beta(Ax^{k+1} + \mathfrak{B}\eta^k - b) = \beta(Ax^{k+1} + \mathfrak{B}\eta^{k+1} - b) + \beta\mathfrak{B}(\eta^k - \eta^{k+1})$, we have

$$\begin{aligned}
&\|v^k - \widetilde{v}^k\|_H^2 - \|\widetilde{v}^k - v^{k+1}\|_H^2 \\
=&\|\eta^k - \widetilde{\eta}^k\|_E^2 + \beta\|\mathfrak{B}(\eta^k - \widetilde{\eta}^k)\|^2 + (2-\gamma)\beta\|A\widetilde{x}^k + \mathfrak{B}\widetilde{\eta}^k - b\|^2 + \\
&2\beta(\eta^k - \eta^{k+1})^T\mathfrak{B}^T(A\widetilde{x}^k + \mathfrak{B}\widetilde{\eta}^k - b).
\end{aligned} \tag{4.1.31}$$

By **Lemma 4.1.2**,

$$\begin{aligned}
&(\eta^k - \eta^{k+1})^T\mathfrak{B}^T\beta(Ax^{k+1} + \mathfrak{B}\eta^{k+1} - b) \\
\geq&(\gamma-1)\beta(\eta^k - \eta^{k+1})^T\mathfrak{B}^T(Ax^k + \mathfrak{B}\eta^k - b) + \frac{1}{2}\|\eta^{k+1} - \eta^k\|_E^2 - \frac{1}{2}\|\eta^k - \eta^{k-1}\|_E^2,
\end{aligned} \tag{4.1.32}$$

thus,

$$\begin{aligned}
&\|v^k - \widetilde{v}^k\|_H^2 - \|\widetilde{v}^k - v^{k+1}\|_H^2 \\
\geq&\|\eta^k - \widetilde{\eta}^k\|_E^2 + \beta\|\mathfrak{B}(\eta^k - \widetilde{\eta}^k)\|^2 + (2-\gamma)\beta\|A\widetilde{x}^k + \mathfrak{B}\widetilde{\eta}^k - b\|^2 + \\
&2(\gamma-1)\beta(\eta^k - \eta^{k+1})^T\mathfrak{B}^T(Ax^k + \mathfrak{B}\eta^k - b) + \|\eta^{k+1} - \eta^k\|_E^2 - \|\eta^k - \eta^{k-1}\|_E^2.
\end{aligned} \tag{4.1.33}$$

Since $2ab \geq -a^2 - b^2$ for $\forall a, b \in \Re$, we have

$$\begin{aligned}
&2(\gamma-1)\beta(\eta^k - \eta^{k+1})^T\mathfrak{B}^T(Ax^k + \mathfrak{B}\eta^k - b) \\
\geq& -[(1-\gamma)^2\beta\|Ax^k + \mathfrak{B}\eta^k - b\|^2 + \beta\|\mathfrak{B}(\eta^k - \eta^{k+1})\|^2].
\end{aligned} \tag{4.1.34}$$

Substituting (4.1.34) into the right hand of (4.1.33), we obtain

$$\begin{aligned}
&\|v^k - \widetilde{v}^k\|_H^2 - \|\widetilde{v}^k - v^{k+1}\|_H^2 \\
\geq&2\|\eta^{k+1} - \eta^k\|_E^2 - \|\eta^{k-1} - \eta^k\|_E^2 + (2-\gamma)\beta\|A\widetilde{x}^k + \mathfrak{B}\widetilde{\eta}^k - b\|^2 - (1-\gamma)^2\beta\|Ax^k + \mathfrak{B}\eta^k - b\|^2.
\end{aligned} \tag{4.1.35}$$

By identity $(1-\gamma)^2 = (1+\gamma-\gamma^2) + (1-\gamma)^2$, we obtain (4.1.28). □

Combining **Lemma 4.1.1** and **Lemma 4.1.3**, we have the following theorem.

Theorem 4.1.2. *Let $w^k = (x^k, \eta^k, \lambda^k)$ be generated by (4.1.3) and $\gamma \in (0, \frac{1+\sqrt{5}}{2})$, we have*

$$\begin{aligned}
&\theta(u) - \theta(\widetilde{u}^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \\
\geq&\frac{1}{2}[\|v - v^{k+1}\|_H^2 + \|\eta^{k+1} - \eta^k\|_E^2 + (1-\gamma)^2\|Ax^{k+1} + \mathfrak{B}\eta^{k+1} - b\|^2] - \\
&\frac{1}{2}[\|v - v^k\|_H^2 + \|\eta^k - \eta^{k-1}\|_E^2 + (1-\gamma)^2\|Ax^k + \mathfrak{B}\eta^k - b\|^2] + \\
&\frac{1}{2}[\|\eta^{k+1} - \eta^k\|_E^2 + (1+\gamma-\gamma^2)\beta\|A\widetilde{x}^k + \mathfrak{B}\widetilde{\eta}^k - b\|^2], \quad \forall w \in \Omega.
\end{aligned} \tag{4.1.36}$$

Theorem 4.1.3. *Let $w^k = (x^k, \eta^k, \lambda^k)$ be generated by (4.1.3) and $\gamma \in (0, \frac{1+\sqrt{5}}{2})$, we have*

$$\begin{aligned}
&\|v^{k+1} - v^*\|_H^2 + \|\eta^{k+1} - \eta^k\|_E^2 + (1-\gamma)^2\|Ax^{k+1} + \mathfrak{B}\eta^{k+1} - b\|^2 \\
\leq&\|v^k - v^*\|_H^2 + \|\eta^k - \eta^{k-1}\|_E^2 + (1-\gamma)^2\|Ax^k + \mathfrak{B}\eta^k - b\|^2 - \\
&(\|\eta^{k+1} - \eta^k\|_E^2 + (1+\gamma-\gamma^2)\beta\|A\widetilde{x}^k + \mathfrak{B}\widetilde{\eta}^k - b\|^2), \quad \forall w^* \in \Omega^*.
\end{aligned} \tag{4.1.37}$$

Proof. Setting $w = w^*$ in (4.1.36) and by optimal condition (2.1.5) (2.1.6) of ADMM for three separable operators, the theorem is established. □

4.2 Convergence Analysis

Theorem 4.1.3 is foundation for convergence analysis of algorithm (4.1.3).

Theorem 4.2.1. *Let $\{w^k\}$ be the sequence generated by iterative algorithm (4.1.3), then each accumulation point of $\{w^k\}$ is the optimal point.*

Proof. By (4.1.37), we have

$$\begin{aligned}&\sum_{k=1}^{\infty}[\|\eta^{k+1}-\eta^k\|_E^2+(1+\gamma-\gamma^2)\beta\|Ax^{k+1}+\mathfrak{B}\eta^{k+1}-b\|^2]\\ \le&\|v^*-v^1\|_H^2+\|\eta^1-\eta^0\|_E^2+(1-\gamma)^2\|Ax^1+\mathfrak{B}\eta^1-b\|^2<\infty.\end{aligned} \tag{4.2.1}$$

Therefore,

$$\lim_{k\to\infty}[\|\eta^{k+1}-\eta^k\|_E^2+(1+\gamma-\gamma^2)\beta\|Ax^{k+1}+\mathfrak{B}\eta^{k+1}-b\|^2]=0. \tag{4.2.2}$$

Since $1+\gamma-\gamma^2>0$ with $\forall\gamma\in(0,\frac{1+\sqrt{5}}{2})$, we have

$$\lim_{k\to\infty}\|\eta^{k+1}-\eta^k\|=0,\quad \lim_{k\to\infty}\|Ax^{k+1}+\mathfrak{B}\eta^{k+1}-b\|=0. \tag{4.2.3}$$

Since

$$\|v^k-v^{k+1}\|_H^2=\|\eta^{k+1}-\eta^k\|_E^2+\beta\|\mathfrak{B}(\eta^{k+1}-\eta^k)\|^2+\gamma\beta\|Ax^{k+1}+\mathfrak{B}\eta^{k+1}-b\|^2, \tag{4.2.4}$$

we have

$$\lim_{k\to\infty}\|v^k-v^{k+1}\|_H^2=0. \tag{4.2.5}$$

By (4.2.1), we have

$$\|v^{k+1}-v^*\|_H^2\le\|v^*-v^1\|_H^2+\|\eta^1-\eta^0\|_E^2+(1-\gamma)^2\|Ax^1+\mathfrak{B}\eta^1-b\|^2, \tag{4.2.6}$$

and

$$(1-\gamma)^2\beta\|Ax^{k+1}+\mathfrak{B}\eta^{k+1}-b\|^2\le\|v^*-v^1\|_H^2+\|\eta^1-\eta^0\|_E^2+(1-\gamma)^2\|Ax^1+\mathfrak{B}\eta^1-b\|^2, \tag{4.2.7}$$

thus, $\{Ax^k\}$ and $\{v^k\}$ are bounded.

With the assumption that A has full column rank, $\{w^k\}$ and $\{\widetilde{w}^k\}$ are bounded, then let $\{\widetilde{w}^{k_j}\}$ be a subsequence of $\{\widetilde{w}^k\}$ such that $\lim_{j\to\infty}\widetilde{w}^{k_j}=w^\infty$, we have

$$\begin{aligned}\theta(u)-\theta(\widetilde{u}^{k_j})+(w-\widetilde{w}^{k_j})^TF(\widetilde{w}^{k_j})&\ge(v-\widetilde{v}^{k_j})^TQ(v^{k_j}-\widetilde{v}^{k_j})\\&\ge(v-\widetilde{v}^{k_j})^TH(v^{k_j}-v^{k_j+1})\\&\ge-\|v-\widetilde{v}^{k_j}\|\|H(v^{k_j}-v^{k_j+1})\|\overset{j\to\infty}{\longrightarrow}0,\end{aligned} \tag{4.2.8}$$

then

$$\theta(u)-\theta(u^\infty)+(w-w^\infty)^TF(w^\infty)\ge0,\quad\forall\omega\in\Omega. \tag{4.2.9}$$

Therefore w^∞ is optimal point of and proof is complete. □

Similar to equalized method, algorithm (4.1.3) has a worst-case $O(\frac{1}{n})$ convergence rate in ergodic sense.

Theorem 4.2.2. *The $\{w^k\}$ generated by iterative algorithm (4.1.3) has $O(\frac{1}{n})$ convergence rate in ergodic sense.*

Proof. By (4.1.36) and (2.1.6), we have

$$\begin{aligned} &\theta(u)-\theta(\widetilde{u}^k)+(w-\widetilde{w}^k)^TF(w)\\ \geq&\frac{1}{2}[\|v-v^{k+1}\|_H^2+\|\eta^{k+1}-\eta^k\|_E^2+(1-\gamma)^2\|Ax^{k+1}+\mathfrak{B}\eta^{k+1}-b\|^2]-\\ &\frac{1}{2}[\|v-v^k\|_H^2+\|\eta^k-\eta^{k-1}\|_E^2+(1-\gamma)^2\|Ax^k+\mathfrak{B}\eta^k-b\|^2],\quad \forall w\in\Omega. \end{aligned} \tag{4.2.10}$$

Let $\widetilde{w}_n^k=\frac{1}{n}\sum_{k=1}^n\widetilde{w}^k$ and $\widetilde{u}_n^k=\frac{1}{n}\sum_{k=1}^n\widetilde{u}^k$.

Summarizing (4.2.10) over $k=1,2,\cdots,n$, we have

$$\begin{aligned} &n\theta(u)-\sum_{k=1}^n\theta(\widetilde{u}^k)+(nw-\sum_{k=1}^n\widetilde{w}^k)^TF(w)\\ \geq&-\frac{1}{2}[\|v-v^1\|_H^2+\|\eta^1-\eta^0\|_E^2+(1-\gamma)^2\|Ax^1+\mathfrak{B}\eta^1-b\|^2],\quad \forall w\in\Omega. \end{aligned} \tag{4.2.11}$$

(4.2.11) can be rewritten as

$$\begin{aligned} &\frac{1}{n}\sum_{k=1}^n\theta(\widetilde{u}^k)-\theta(u)+(\widetilde{w}_n^k-w)^TF(w)\\ \leq&\frac{1}{2n}[\|v-v^1\|_H^2+\|\eta^1-\eta^0\|_E^2+(1-\gamma)^2\|Ax^1+\mathfrak{B}\eta^1-b\|^2],\quad \forall w\in\Omega. \end{aligned} \tag{4.2.12}$$

Since $\theta(\widetilde{u}_n^k)\leq\frac{1}{n}\sum_{k=1}^n\theta(\widetilde{u}^k)$, we have

$$\begin{aligned} &\theta(\widetilde{u}_n^k)-\theta(u)+(\widetilde{w}_n^k-w)^TF(w)\\ \leq&\frac{1}{2n}[\|v-v^1\|_H^2+\|\eta^1-\eta^0\|_E^2+(1-\gamma)^2\|Ax^1+\mathfrak{B}\eta^1-b\|^2],\quad \forall w\in\Omega. \end{aligned} \tag{4.2.13}$$

Recalling the definition of ϵ-approximation solution in chapter 2, we have proven the theorem by (4.2.13). □

Remark 4.2.1. *If we let γ be equal to 1, then algorithm (4.1.3) is equalized method introduced in chapter 3.*

4.3 Accelerated Variant of Equalized Method

As mentioned in chapter 3, there is a variant of equalized method which equalizes x,y-step not x,z-step by adding regularization term. Since x^k participates in following iteration, this variant is not ADMM-like. We can also add a relaxation factor γ on λ-step, that is,

$$\begin{cases} x^{k+1}=\arg\min\{L_\beta(x,y^k,z^k,\lambda^k)+\frac{\tau\beta}{2}\|A(x-x^k)\|^2|x\in\mathcal{X}\},\\ y^{k+1}=\arg\min\{L_\beta(x^k,y,z^k,\lambda^k)+\frac{\tau\beta}{2}\|B(y-y^k)\|^2|y\in\mathcal{Y}\},\\ z^{k+1}=\arg\min\{L_\beta(x^{k+1},y^{k+1},z,\lambda^k)|z\in\mathcal{Z}\},\\ \lambda^{k+1}=\lambda^k-\gamma\beta(Ax^{k+1}+By^{k+1}+Cz^{k+1}-b), \end{cases} \tag{4.3.1}$$

where $\tau>1$ and $\gamma\in(0,\frac{1+\sqrt{5}}{2})$.

The same results of convergence are also established and the detailed proof is omitted because it is similar to that of equalized method with relaxation factor.

Chapter 5 Numerical Results

In this chapter, three examples will be illustrated to verify the effectiveness and efficiency of algorithms introduced in previous chapters. The corresponding codes are attached in Appendix B.

5.1 Solving a Counterexample

In [11], we have known that the direct extension of ADMM may not be convergent since a counterexample is provided. The example is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^2 + \frac{1}{2}y^2 + \frac{1}{2}z^2 \\ \text{subject to} & Ax + By + Cz = 0 \end{array} \tag{5.1.1}$$

where

$$(A, B, C) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix}. \tag{5.1.2}$$

Obviously, the optimal point of problem(5.1.1) is $(0, 0, 0)$ and optimal value is 0. We use the direct extension method to solve it, then we have following iterative equation

$$v^{k+1} = Mv^k = \cdots = M^{k+1}v^0, \tag{5.1.3}$$

where $v = (y, z, \lambda)$ and M is a matrix whose spectral radius $\rho(M) > 1$. Therefore, we can choose a nonzero v^0 such that $\lim_{k\to\infty}\|v^k\| = +\infty$.

If we use methods in chapter 3 and chapter 4 to solve problem(5.1.1), then we can obtain the answer without difficulty. The following results are obtained with initial point $(x, y, z) = (1, 1, 1)$, fixed error bound $STOL = 10^{-4}$ and fixed parameter $\beta = 1$, and show how objective function value varies with iteration in three methods.

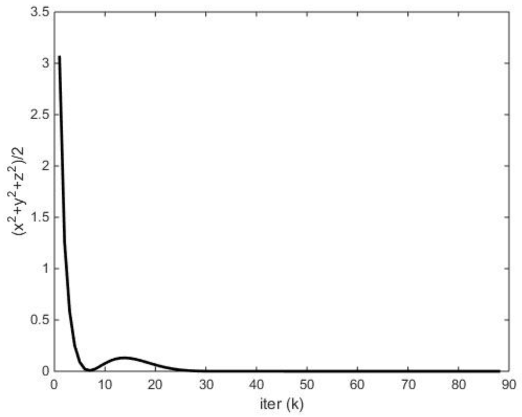

Figure 5.1: Corrected method for counterexample with $\nu = 0.9$.

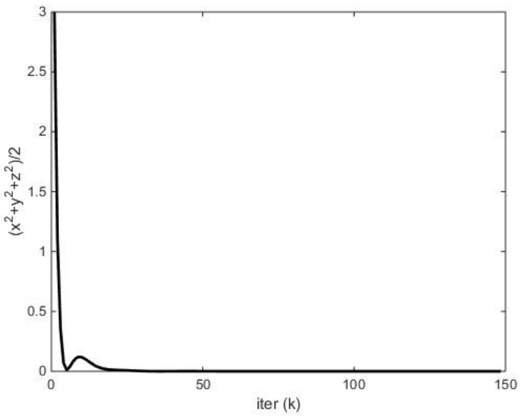

Figure 5.2: Equalized method for counterexample with $\tau = 1.1$.

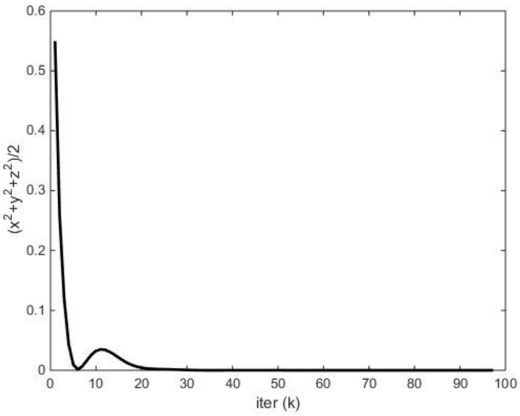

Figure 5.3: Variant of equalized method for counterexample with $\tau = 1.1$.

As illustrated in figure 5.1, 5.2 and,5.3, this kind of method which iterates x, y and z-subproblem alternately will approach optimal value in very few steps. If we don't require the answer in very high degree of accuracy, then these algorithms can become more efficiency.

5.2 Lasso with Elastic Net

Lasso(Least absolute shrinkage and selection operator) is stated as

$$\text{minimize} \quad \|Kx - b\|_2^2 + \lambda\|x\|_1 \tag{5.2.1}$$

where $K \in \Re^{m\times n}$, $x \in \Re^n$, $b \in \Re^m$ and $\lambda \in \Re$.

This method is firstly introduced by Robert Tibshirani in 1996. Since l_1-norm has shrinkage property, Lasso can select variables and generate sparse model by adjusting penalty parameter λ. It is also closely related to BPDN(basis pursuit denoising). However, Lasso has some limitations:

1.If $n >> m$, the lasso selects at most m variables. The number of selected variables is bounded by the number of samples.

2.Grouped variables: the lasso fails to do grouped selection. It tends to select one variable from a group and ignore the others.

Thus, Lasso with elastic net is introduced by Zou and Hastie in 2005 to address shortcomings of Lasso, that is,

$$\text{minimize} \quad \|Kx - b\|_2 + \lambda_2\|x\|_2 + \lambda_1\|x\|_1 \tag{5.2.2}$$

where $K \in \Re^{m\times n}$, $x \in \Re^n$, $b \in \Re^m$ and $\lambda_1, \lambda_2 \in \Re$.

Elastic net regularization encourages grouping effect and stabilizes the l_1 regularization path by adding a l_2-norm.

We can solve problem (5.2.2) by rewriting it in equivalent form,

$$\begin{aligned} &\text{minimize} && \|Kx - b\|_2^2 + \lambda_2\|y\|_2^2 + \lambda_1\|z\|_1 \\ &\text{subject to} && Ax + By + Cz = 0 \end{aligned} \tag{5.2.3}$$

where $A = (I_n, I_n)^T$, $B = (-I_n, 0)^T$ and $C = (0, -I_n)^T$.

The following table shows the number of iteration steps and the iteration time in different methods. We let $m = n = N$, $\lambda_1 = 1$, $\lambda_2 = 1$, $\beta = 1$ and $STOL = 10^{-3}$.

Table 5.1: Description of various methods

Methods	Description
Method 1	Corrected method, $\nu = 0.9$
Method 2	Equalized method, $\tau = 1.1$
Method 3	Variant of equalized method, $\tau = 1.1$
Method 4	Equalized method with relaxation factor, $\tau = 1.1$, $\gamma = 1.5$
Method 5	Variant of equalized method with relaxation factor, $\tau = 1.1$, $\gamma = 1.5$

Table 5.2: The results for problem with matrix $K \in \Re^{N\times N}$ in different methods

	Method 1		Method 2		Method 3		Method 4		Method 5	
N	Iter	T[sec]	Iter	T[sec]	Iter	T[sec]	Iter	T[sec]	Iter	T[sec]
100	40	0.11	38	0.22	35	0.36	25	0.20	25	0.12
200	46	0.41	45	0.68	42	0.53	30	0.41	28	0.32
500	55	3.34	54	7.41	51	6.37	36	4.94	34	4.06
1000	59	22.62	58	51.60	56	46.83	40	39.80	37	31.00
1500	62	56.50	61	177.44	59	155.93	41	127.13	39	103.79
2000	62	119.57	61	383.85	59	359.74	42	254.54	39	245.25

5.3 Lasso with Constraints

We also provide another example about Lasso with constraints. The problem is stated as

$$\begin{aligned} &\text{minimize} \quad \|Kx - b\|_2^2 + \lambda_1 \|x\|_1 \\ &\text{subject to} \quad x \in \Re^n_+ \end{aligned} \tag{5.3.1}$$

where $K \in \Re^{m\times n}$, $x \in \Re^n$, $b \in \Re^m$ and $x \in \Re^n_+$ means each component of x is no less than 0.

We can also reformulate it into an equivalent form,

$$\begin{aligned} &\text{minimize} \quad \|Kx - b\|_2^2 + \lambda_1 \|y\|_1 + \chi_{\Re^n_+}(z) \\ &\text{subject to} \quad Ax + By + Cz = 0 \end{aligned} \tag{5.3.2}$$

where $A = (I_n, I_n)^T$, $B = (-I_n, 0)^T$ and $C = (0, -I_n)^T$, and

$$\chi_{\Re^n_+}(z) = \begin{cases} 0, & z \in \Re^n_+, \\ +\infty, & otherwise. \end{cases} \tag{5.3.3}$$

We also let $m = n = N$, $\lambda_1 = 1$, $\beta = 1$, $STOL = 10^{-3}$, and show the iteration time and the number of iteration steps in the following table.

Table 5.3: Description of various methods

Methods	Description
Method 1	Corrected method, $\nu = 0.9$
Method 2	Equalized method, $\tau = 1.1$
Method 3	Variant of equalized method, $\tau = 1.1$
Method 4	Equalized method with relaxation factor, $\tau = 1.1$, $\gamma = 1.5$
Method 5	Variant of equalized method with relaxation factor, $\tau = 1.1$, $\gamma = 1.5$

Table 5.4: The results for problem with matrix $K \in \Re^{N\times N}$ in different methods

	Method 1		Method 2		Method 3		Method 4		Method 5	
N	Iter	T[sec]	Iter	T[sec]	Iter	T[sec]	Iter	T[sec]	Iter	T[sec]
100	27	0.07	38	0.07	46	0.09	34	0.06	47	0.09
200	31	0.17	41	0.19	47	0.30	35	0.18	48	0.30
500	34	1.23	43	1.48	38	2.02	30	1.01	37	1.96
1000	42	8.65	49	9.62	38	12.58	33	6.48	30	10.51
1500	44	26.95	51	32.11	40	40.91	34	20.28	30	31.42
2000	43	62.14	49	66.59	39	93.64	34	46.15	29	69.60

All codes are written in Matlab and run on a personal laptop. Clearly, CPU time is not precise, but is enough to compare different methods. According to table 5.2 and table 5.4, accelerated algorithms can prominently shorten the time and the number of steps which iteration needs.

Conclusions

In this paper, we provide two improved algorithms based prediction-correction algorithm framework to correct the direct extension of ADMM for three separable operators. A derivative algorithm from equalized method is also mentioned and we have completely proved convergence property of equalized method with relaxation factor which can accelerate original algorithm. Only three separable operators situation and part of results are discussed, and the whole discussion is based on variational inequality which is easy to understand. These algorithms are efficient to solve some large-scale convex optimization problems with separable structure. There are some open problems in these area for future study:

1. What other conditions should convex optimization problems satisfy to be convergent in the direct extension of ADMM for three separable operators?

2. Is self-adaptive penalty parameter method also available for the direct extension of ADMM for three separable operators?

References

[1] Rockafellar R T. Convex analysis. Princeton University Press, 1970.

[2] Boyd S, Vandenberghe L. Convex optimization. Cambridge University Press, 2004.

[3] Facchinei F, Pang J S. Finite-dimensional variational inequalities and complementarity problems, volume Vol. I. Springer Ser. Oper. Res., Springer-Verlag, New York, 2003.

[4] Rockafellar R T, Wets R J B. Variational analysis, volume Volume 317. Springer, 1998.

[5] Glowinski, Marrocco. Sur lapproximation par elements finis dordre un et la resolution par penalisation-dualite d'une classe de problemes de Dirichlet non lineaires. Rech. Oper., Anal. Numer., 1975, 2:41–76.

[6] Glowinski. Numerical mehtods for nonlinear variational problems. Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[7] Rockafellar. Monotone operators and the proximal point algorithm. SIAM J. Cont. Optim., 1976, 14:877–898.

[8] Martinet. Regularisation, dinequations variationelles par approximations succesives. Recherche Oper., 1970, 4:154–159.

[9] He B, Yang H, Wang. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. Journal of Optimization Theory and Applications, 2000, 106:337–356.

[10] He B, Yang H. Some convergence properties of a method of multipliers for linearly constarined monotone variational inequalities. Operations Research Letters, 1998, 23:151–161.

[11] Chen C, He B, Yuan X. The direct extension of ADMM for multi-block convex minimization problems is not necessary convergent. Mathematical Programming, 2016, 155:57–79.

[12] He B, Yang H. Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities. Operations Reaserch Letters, 1998, 23:151–161.

[13] Nagurney A, Ramanujam P. Transportation network policy modeling with goal targets and generalized penalty functions. Transportion Sci., 1996, 30:3–13.

[14] Li L. Selected Applications of Convex Optimization. Springer, 2015.

[15] Cai X, He B. A proximal point algorithm revisit on the alternating direction method of multipliers. Science China Mathematics, 2013, 56:2179–2186.

[16] Boyd S. Distirbuted optimization and statistical learning via the alternating direction method of multipliers. Foun. Trends Mach. Learn., 2010, 3:1–122.

[17] 何炳生. 生活理念对设计优化分裂算法的帮助—以改造 ADMM 求解三个可分离算子问题为例. 见 http://maths.nju.edu.cn/ hebma/中的《My Talk》.

[18] He B, Tao M, Yuan X. Alternating direction method with Gaussian back substitution for separable convex programming. SIAM Journal on Optimization, 2012, 22:313–340.

[19] He B, Tao M, Yuan X. A splitting method for separable convex programming. IMA Journal of Numerical Analysis, 2015, 31:394–426.

[20] He B, Yuan X. On the O(1/n) convergence rate of the alternating direction method. SIAM J. Numerical Analysis, 2012, 50:700–709.

Appendix A Convergence Analysis of ADMM with Relaxation Factor

A.1 Optimal Condition of ADMM

As introduced in background of chapter 1, ADMM(alternating direction method of multipliers) aims at the following problem

$$\begin{array}{ll} \text{minimize} & \theta_1(x) + \theta_2(y) \\ \text{subject to} & Ax + By = b \end{array} \tag{A.1.1}$$

where $x \in \mathcal{X} \subset \Re^{n_1}$, $y \in \mathcal{Y} \subset \Re^{n_2}$, $A \in \Re^{m \times n_1}$, $B \in \Re^{m \times n_2}$, $b \in \Re^m$ and θ_1, θ_2 are convex and $\mathcal{X}, \mathcal{Y}$ are closed, convex.

According to adding a relaxation factor on λ-step, ADMM with relaxation factor is formed as

$$\begin{cases} x^{k+1} = \arg\min\{L_\beta(x, y^k, \lambda^k) | x \in \mathcal{X}\}, \\ y^{k+1} = \arg\min\{L_\beta(x^{k+1}, y, \lambda^k) | y \in \mathcal{Y}\}, \\ \lambda^{k+1} = \lambda^k - \gamma\beta(Ax^{k+1} + By^{k+1} - b), \end{cases} \tag{A.1.2}$$

where $\gamma \in (0, \frac{1+\sqrt{5}}{2})$.

By **Theorem 1.1.1**, we also have following saddle inequalities

$$L_{\lambda \in \Re^m}(x^*, y^*, \lambda) \leq L(x^*, y^*, \lambda^*) \leq L_{(x,y) \in \mathcal{X} \times \mathcal{Y}}(x, y, \lambda^*), \tag{A.1.3}$$

where $(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$ is optimal point of problem(A.1.1) and $\lambda^* \in \Re^m$ is optimal point of Lagrangian dual function.

Thus, we can have optimal conditions of problem (A.1.1) in form of variational inequality,

$$\begin{cases} x^* \in \mathcal{X}, & L(x, y^*, \lambda) - L(x^*, y^*, \lambda^*) \geq 0, \quad \forall x \in \mathcal{X}, \\ y^* \in \mathcal{Y}, & L(x^*, y, \lambda) - L(x^*, y^*, \lambda^*) \geq 0, \quad \forall y \in \mathcal{Y}, \\ \lambda^* \in \Re^m, & L(x^*, y^*, \lambda^*) - L(x^*, y^*, \lambda) \geq 0, \quad \forall \lambda \in \Re^m. \end{cases} \tag{A.1.4}$$

Rewriting (A.1.4) into a compact form, we have

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall w \in \Omega, \tag{A.1.5}$$

where

$$w = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, \quad u = \begin{pmatrix} x \\ y \end{pmatrix}, \quad F(w) = \begin{pmatrix} -A^T\lambda \\ -B^T\lambda \\ Ax + By - b \end{pmatrix},$$

$$\theta(u) = \theta_1(x) + \theta_2(y), \quad \text{and} \quad \Omega = \mathcal{X} \times \mathcal{Y} \times \Re^m,$$

and since F is skew-symmetric, we also have

$$(w - \bar{w})^T (F(w) - F(\bar{w})) = 0, \quad \forall w, \bar{w} \in \Omega. \tag{A.1.6}$$

A.2 Convergence Analysis

By **Lemma 1.1.1**, expanding (A.1.2), we get

$$\begin{cases} \theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^T[-A^T\lambda^k + \beta A^T(Ax^{k+1} + By^k - b)] \geq 0, \\ \theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^T[-B^T\lambda^k + \beta B^T(Ax^{k+1} + By^{k+1} - b)] \geq 0, \\ (Ax^{k+1} + By^{k+1} - b) - \dfrac{1}{\gamma\beta}(\lambda^k - \lambda^{k+1}) = 0. \end{cases} \tag{A.2.1}$$

We can consider (A.2.1) in prediction-correction algorithm framework, and we define predictive variables as $\widetilde{w}^k = (\widetilde{x}^k, \widetilde{y}^k, \widetilde{\lambda}^k)$, where

$$(\widetilde{x}^k, \widetilde{y}^k) = (x^{k+1}, y^{k+1}), \tag{A.2.2}$$

and

$$\widetilde{\lambda}^k = \lambda^k - \beta(A\widetilde{x}^k + By^k - b). \tag{A.2.3}$$

Rewriting (A.2.1), we get

$$\theta(u) - \theta(\widetilde{u}^k) + (w - \widetilde{w}^k)^T F(\widetilde{w}^k) \geq (v - \widetilde{v}^k)^T Q_0 (v^k - \widetilde{v}^k), \tag{A.2.4}$$

where

$$Q_0 = \begin{pmatrix} \beta B^T B & 0 \\ -B & \frac{1}{\beta} I \end{pmatrix}. \tag{A.2.5}$$

Then defining corrective variables as

$$\begin{aligned} M_0 &= \begin{pmatrix} I & 0 \\ -\gamma\beta B & \frac{1}{\gamma} I \end{pmatrix}, \quad w^{k+1} = w^k - M(w^k - \widetilde{w}^k), \\ H_0 &= Q_0 M_0^{-1} = \begin{pmatrix} \beta B^T B & 0 \\ 0 & \frac{1}{\gamma\beta} I \end{pmatrix} \succ 0, \end{aligned} \tag{A.2.6}$$

and by **Proposition 1.1.1**, we have identity

$$\begin{aligned} (v - \widetilde{v}^k)^T Q_0 (v^k - \widetilde{v}^k) &= (v - \widetilde{v}^k)^T H_0 (v^k - v^{k+1}) \\ &= \frac{1}{2}\|v - v^{k+1}\|_{H_0}^2 - \frac{1}{2}\|v - v^k\|_{H_0}^2 + \frac{1}{2}\|\widetilde{v}^k - v^k\|_{H_0}^2 - \frac{1}{2}\|\widetilde{v}^k - v^{k+1}\|_{H_0}^2. \end{aligned} \tag{A.2.7}$$

Noticing that the $\|\widetilde{v}^k - v^k\|_{H_0}^2 - \|\widetilde{v}^k - v^{k+1}\|_{H_0}^2$ can be rewritten as

$$\begin{aligned} &\|\widetilde{v}^k - v^k\|_{H_0}^2 - \|\widetilde{v}^k - v^{k+1}\|_{H_0}^2 \\ &= \beta\|B(y^k - \widetilde{y}^k)\|^2 + \frac{1}{\beta\gamma}\|\lambda^k - \widetilde{\lambda}^k\|^2 - \frac{1}{\beta\gamma}\|\lambda^k - \widetilde{\lambda}^k - \beta\gamma(A\widetilde{x}^k + B\widetilde{y}^k - b)\|^2 \\ &= \beta\|B(y^k - \widetilde{y}^k)\|^2 + 2(\lambda^k - \widetilde{\lambda}^k)^T(A\widetilde{x}^k + B\widetilde{y}^k - b) - \beta\gamma\|A\widetilde{x}^k + B\widetilde{y}^k - b\|^2 \\ &= \beta\|B(y^k - \widetilde{y}^k)\|^2 + (2-\gamma)\beta\|A\widetilde{x}^k + B\widetilde{y}^k - b\|^2 + 2\beta(y^k - y^{k+1})^T B^T(A\widetilde{x}^k + B\widetilde{y}^k - b). \end{aligned} \tag{A.2.8}$$

Lemma A.2.1. *Let (x^k,y^k,λ^k) be generated by algorithm (A.1.2) above, then it satisfies*

$$\begin{aligned} &\beta(y^k-y^{k+1})^TB^T(A\widetilde{x}^k+B\widetilde{y}^k-b)\\ \ge\ &(y^k-y^{k+1})^TB^T(\lambda^k-\lambda^{k-1})+\beta(y^k-y^{k+1})^TB^T(Ax^k+By^k-b). \end{aligned} \tag{A.2.9}$$

Proof. By y-step in (A.2.1), we have

$$\begin{cases} \theta_2(y^k)-\theta_2(y^{k+1})+(y^k-y^{k+1})^T[-B^T\lambda^k+\beta B^T(Ax^{k+1}+By^{k+1}-b)]\ge 0,\\ \theta_2(y^{k+1})-\theta_2(y^k)+(y^{k+1}-y^k)^T[-B^T\lambda^{k-1}+\beta B^T(Ax^k+By^k-b)]\ge 0. \end{cases} \tag{A.2.10}$$

□

Adding two equations in (A.1.16) and the lemma is proved.

Now we can prove the following theorem.

Theorem A.2.1. *Assume A and B have full column rank, let $\{w^k\}$ be sequence generated by algorithm (A.1.2), then each accumulation point of $\{w^k\}$ is an optimal point.*

Proof. By (A.2.8) and **Lemma A.2.1**, we have

$$\begin{aligned} &\|\widetilde{v}^k-v^k\|_{H_0}^2-\|\widetilde{v}^k-v^{k+1}\|_{H_0}^2\\ \ge\ &\beta\|B(y^k-\widetilde{y}^k)\|^2+(2-\gamma)\beta\|A\widetilde{x}^k+B\widetilde{y}^k-b\|^2+2(y^k-y^{k+1})^TB^T(\lambda^k-\lambda^{k-1})+\\ &2\beta(y^k-y^{k+1})^TB^T(Ax^k+By^k-b)\\ \ge\ &\beta\|B(y^k-\widetilde{y}^k)\|^2+(2-\gamma)\beta\|A\widetilde{x}^k+B\widetilde{y}^k-b\|^2+\\ &2(1-\gamma)\beta(y^k-y^{k+1})^TB^T(Ax^k+By^k-b). \end{aligned} \tag{A.2.11}$$

We discuss γ in two cases.

(1) $0<\gamma\le 1:$

Since

$$\begin{aligned} &2(1-\gamma)\beta(y^k-y^{k+1})^TB^T(Ax^k+By^k-b)\\ \ge\ &-(1-\gamma)\beta\|B(y^k-y^{k+1})\|^2-(1-\gamma)\beta\|Ax^k+By^k-b\|^2, \end{aligned} \tag{A.2.12}$$

we have

$$\begin{aligned} \|\widetilde{v}^k-v^k\|_{H_0}^2-\|\widetilde{v}^k-v^{k+1}\|_{H_0}^2\ \ge\ &\gamma\beta\|B(y^k-y^{k+1})\|^2+\\ &(2-\gamma)\beta\|Ax^{k+1}+By^{k+1}-b\|^2-(1-\gamma)\beta\|Ax^k+By^k-b\|^2. \end{aligned} \tag{A.2.13}$$

By the right hand of (A.2.7) and (A.2.13), we have

$$\begin{aligned} &\frac{1}{2}\|v-v^{k+1}\|_{H_0}^2-\frac{1}{2}\|v-v^k\|_{H_0}^2+\frac{1}{2}\|\widetilde{v}^k-v^k\|_{H_0}^2-\frac{1}{2}\|\widetilde{v}^k-v^{k+1}\|_{H_0}^2\\ \ge\ &\left[\frac{1}{2}\|v-v^{k+1}\|_{H_0}^2+\frac{1}{2}(1-\gamma)\beta\|Ax^{k+1}+By^{k+1}-b\|^2\right]-\\ &\left[\frac{1}{2}\|v-v^k\|_{H_0}^2+\frac{1}{2}(1-\gamma)\beta\|Ax^k+By^k-b\|^2\right]+\\ &\left[\frac{1}{2}\gamma\beta\|B(y^k-y^{k+1})\|^2+\frac{1}{2}\beta\|Ax^{k+1}+By^{k+1}-b\|^2\right]. \end{aligned} \tag{A.2.14}$$

Substituting (A.2.14) to the right hand of (A.2.7) and according to optimal condition (A.1.5), we obtain

$$\begin{aligned}
&\|v^{k+1}-v^*\|_{H_0}^2+(1-\gamma)\beta\|Ax^{k+1}+By^{k+1}-b\|^2\\
\le&\|v^k-v^*\|_{H_0}^2+(1-\gamma)\beta\|Ax^k+By^k-b\|^2-\\
&\left[\gamma\beta\|B(y^k-y^{k+1})\|^2+\beta\|Ax^{k+1}+By^{k+1}-b\|^2\right].
\end{aligned} \tag{A.2.15}$$

(2) $1\ge\gamma<\frac{1+\sqrt{5}}{2}$:

Since

$$\begin{aligned}
&2(1-\gamma)\beta(y^k-y^{k+1})^TB^T(Ax^k+By^k-b)\\
\ge&-\gamma(\gamma-1)\beta\|B(y^k-y^{k+1})\|^2-\frac{\gamma-1}{\gamma}\beta\|Ax^k+By^k-b\|^2,
\end{aligned} \tag{A.2.16}$$

we have

$$\begin{aligned}
\|\widetilde{v}^k-v^k\|_{H_0}^2-\|\widetilde{v}^k-v^{k+1}\|_{H_0}^2\ \ge\ &(1+\gamma-\gamma^2)\beta\|B(y^k-\widetilde{y}^k)\|^2+\\
&(2-\gamma)\beta\|Ax^{k+1}+By^{k+1}-b\|^2-\frac{\gamma-1}{\gamma}\beta\|Ax^k+By^k-b\|^2.
\end{aligned} \tag{A.2.17}$$

By the right hand of (A.2.7) and (A.2.17), we obtain

$$\begin{aligned}
&\frac{1}{2}\|v-v^{k+1}\|_{H_0}^2-\frac{1}{2}\|v-v^k\|_{H_0}^2+\frac{1}{2}\|\widetilde{v}^k-v^k\|_{H_0}^2-\frac{1}{2}\|\widetilde{v}^k-v^{k+1}\|_{H_0}^2\\
\ge&\left[\frac{1}{2}\|v-v^{k+1}\|_{H_0}^2+\frac{1}{2}(\gamma-1)\beta\|Ax^{k+1}+By^{k+1}-b\|^2\right]-\\
&\left[\frac{1}{2}\|v-v^k\|_{H_0}^2+\frac{1}{2}(\gamma-1)\beta\|Ax^k+By^k-b\|^2\right]+\\
&\left[\frac{1}{2}(1+\gamma-\gamma^2)\beta\|B(y^k-y^{k+1})\|^2+\frac{1}{2}\frac{1+\gamma-\gamma^2}{\gamma}\beta\|Ax^{k+1}+By^{k+1}-b\|^2\right].
\end{aligned} \tag{A.2.18}$$

Substituting (A.2.18) to the right hand of (A.2.7) and by optimal condition (A.1.5), we have

$$\begin{aligned}
&\|v^{k+1}-v^*\|_{H_0}^2+(\gamma-1)\beta\|Ax^{k+1}+By^{k+1}-b\|^2\\
\le&\|v^k-v^*\|_{H_0}^2+(\gamma-1)\beta\|Ax^k+By^k-b\|^2-\\
&\left[(1+\gamma-\gamma^2)\beta\|B(y^k-y^{k+1})\|^2+\frac{1+\gamma-\gamma^2}{\gamma}\beta\|Ax^{k+1}+By^{k+1}-b\|^2\right].
\end{aligned} \tag{A.2.19}$$

We only need to prove the convergence when $1\le\gamma<\frac{1+\sqrt{5}}{2}$ since $0<\gamma\le 1$ is similar. Adding (A.2.19) over $k=0,1,\cdots$, we have

$$\begin{aligned}
&\sum_{k=0}^{\infty}\left[(1+\gamma-\gamma^2)\beta\|B(y^k-y^{k+1})\|^2+\frac{1+\gamma-\gamma^2}{\gamma}\beta\|Ax^{k+1}+By^{k+1}-b\|^2\right]\\
\le&\|v^0-v^*\|_{H_0}^2+(\gamma-1)\beta\|Ax^0+By^0-b\|^2.
\end{aligned} \tag{A.2.20}$$

(A.2.20) indicates that

$$\lim_{k\to\infty}(1+\gamma-\gamma^2)\beta\|B(y^k-y^{k+1})\|^2+\frac{1+\gamma-\gamma^2}{\gamma}\beta\|Ax^{k+1}+By^{k+1}-b\|^2=0, \tag{A.2.21}$$

furthermore,

$$\lim_{k\to\infty}\|v^k-v^{k+1}\|_{H_0}=0. \tag{A.2.22}$$

Also, by (A.2.19), we have

$$\begin{aligned}&\|v^{k+1}-v^*\|_{H_0}^2+(\gamma-1)\beta\|Ax^{k+1}+By^{k+1}-b\|^2\\ \le&\|v^0-v^*\|_{H_0}^2+(\gamma-1)\beta\|Ax^0+By^0-b\|^2.\end{aligned} \tag{A.2.23}$$

Therefore, assume A, B have full column rank, $\{w^k\}$ is bounded. Let $\{w^{k_j}\}$ be a subsequence such that $\lim_{j\to\infty} w^{k_j}=w^\infty$, then

$$\begin{aligned}&\lim_{j\to\infty}\theta(u)-\theta(u^{k_j})+(w-w^{k_j})^TF(w^{k_j})\\ \ge&\lim_{j\to\infty}(v-v^{k_j})^TH_0(v^k-v^{k_j+1})\\ \ge&\lim_{j\to\infty}-\|v-v^{k_j}\|-\|v^k-v^{k_j+1}\|_{H_0}=0,\quad\forall w\in\Omega.\end{aligned} \tag{A.2.24}$$

Therefore, w^∞ is an optimal point. The proof is complete. □

Appendix B Matlab Codes

B.1 Solving a Counterexample

Matlab Code B.1: Corrected method for counterexample

```
clc;
close all;
clear all;

%% Global constants and defaults
A=[1;1;1];
B=[1;1;2];
C=[1;2;2];
t_start = tic;
QUIET = 0;
MAX_ITER = 1000;
beta = 1;
nu = 0.9;
STOL = 1e-4;
%% Data preprocessing
[m,n]=size(A);
ATA=(A')*A;
BTB=(B')*B;
CTC=(C')*C;
AT=A';
BT=B';
CT=C';
M=[eye(n),-(BT*B)\(BT*C);0,eye(n)];
%% Solver
x_k=1;
y_k=1;
z_k=1;
lambda_k=zeros(m,1);
if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\t%10s\t%10s\t%10s\n', ...
    'iter','r norm', 's norm', 'objective', 'x', 'y', 'z');
end

for i=1:MAX_ITER
    v_old=[y_k;z_k];
    %prediction:
    x_k=(1+beta*ATA)\(AT*lambda_k-beta*AT*B*y_k-beta*AT*C*z_k);
    lambda_p=lambda_k-beta*(A*x_k+B*y_k+C*z_k);
    y_k=(1+beta*BTB)\(BT*lambda_k-beta*BT*A*x_k-beta*BT*C*z_k);
    z_k=(1+beta*CTC)\(CT*lambda_k-beta*CT*A*x_k-beta*CT*B*y_k);
    lambda_k=lambda_k-beta*(A*x_k+B*y_k+C*z_k);
    %correction
    v_k=[y_k;z_k];
    v_k=v_old-nu*M*(v_old-v_k);
    y_k=v_k(1);
    z_k=v_k(2);

    % diagnostics, reporting, termination checks
    history.x(i)=x_k;
    history.y(i)=y_k;
    history.z(i)=z_k;
    history.objval(i)=(x_k^2+y_k^2+z_k^2)/2;
    history.r_norm(i)=norm(A*x_k+B*y_k+C*z_k);
    history.s_norm(i)=norm(norm(v_k-v_old), ...
                      norm(lambda_k-lambda_p));
```

```
    history.stol(i)=STOL;
     if ~QUIET
        fprintf('%3d\t\t%10.4f\t%10.4f\t%10.4f\t%10.4f\t
        %10.4f\t%10.4f\n', i, history.r_norm(i),...
        history.s_norm(i), history.objval(i),...
        history.x(i),history.y(i),history.z(i));
    end

    if (history.r_norm(i) < STOL && history.s_norm(i) < STOL)
         break;
    end
end
if ~QUIET
    toc(t_start);
    subplot(2,2,1);
    stem(history.x);
    ylabel('x'); xlabel('iter (k)');
    subplot(2,2,2);
    stem(history.y);
    ylabel('y'); xlabel('iter (k)');
    subplot(2,2,3);
    stem(history.z);
    ylabel('z'); xlabel('iter (k)');
    subplot(2,2,4);
    stem(history.objval)
    ylabel('(x^2+y^2+z^2)/2'); xlabel('iter (k)');

    K = length(history.objval);

    h = figure;
    plot(1:K, history.objval, 'k',...
        'MarkerSize', 10, 'LineWidth', 2);
    ylabel('(x^2+y^2+z^2)/2'); xlabel('iter (k)');

   g = figure;
    subplot(2,1,1);
    semilogy(1:K, max(1e-8, history.r_norm), 'k',...
        1:K, history.stol, 'k--',  'LineWidth', 2);
    ylabel('||r||_2');

    subplot(2,1,2);
    semilogy(1:K, max(1e-8, history.s_norm), 'k',...
        1:K, history.stol, 'k--', 'LineWidth', 2);
    ylabel('||s||_2'); xlabel('iter (k)');
end
```

Matlab Code B.2: Equalized method for counterexample

```
clc;
close all;
clear all;

%% Global constants and defaults
A=[1;1;1];
B=[1;1;2];
C=[1;2;2];
t_start = tic;
QUIET = 0;
MAX_ITER = 1000;
beta = 1;
tau = 1.1;
gamma = 1;
STOL = 1e-4;
```

```
%% Data preprocessing
[m, n]=size(A);
ATA=(A')*A;
BTB=(B')*B;
CTC=(C')*C;
AT=A';
BT=B';
CT=C';
%% Solver
x_k=1;
y_k=1;
z_k=1;
lambda_k=zeros(m,1);
if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\t%10s\t%10s\t%10s\n',...
    'iter','r norm', 's norm', 'objective', 'x', 'y', 'z');
end

for i=1:MAX_ITER
    y_old=y_k;
    x_k=(1+beta*ATA)\(AT*lambda_k-beta*AT*B*y_k-beta*AT*C*z_k);
    lambda_p=lambda_k-beta*(A*x_k+B*y_k+C*z_k);
    y_k=(1+(tau+1)*beta*BTB)\(BT*lambda_k-beta*BT*A*x_k-...
        beta*BT*C*z_k+beta*tau*BTB*y_k);
    z_k=(1+(tau+1)*beta*CTC)\(CT*lambda_k-beta*CT*A*x_k-...
        beta*CT*B*y_old+beta*tau*CTC*z_k);
    lambda_k=lambda_k-gamma*beta*(A*x_k+B*y_k+C*z_k);

    % diagnostics, reporting, termination checks
    history.x(i)=x_k;
    history.y(i)=y_k;
    history.z(i)=z_k;
    history.objval(i)=(x_k^2+y_k^2+z_k^2)/2;
    history.r_norm(i)=norm(A*x_k+B*y_k+C*z_k);
    history.s_norm(i)=norm(lambda_k-lambda_p);
    history.stol(i)=STOL;
    if ~QUIET
        fprintf('%3d\t\t%10.4f\t%10.4f\t%10.4f\t%10.4f\t
                %10.4f\t%10.4f\n', i, history.r_norm(i),...
                history.s_norm(i), history.objval(i),...
                history.x(i),history.y(i),history.z(i));
    end

    if (history.r_norm(i) < STOL && history.s_norm(i) < STOL)
        break;
    end
end
if ~QUIET
    toc(t_start);
    subplot(2,2,1);
    stem(history.x);
    ylabel('x'); xlabel('iter (k)');
    subplot(2,2,2);
    stem(history.y);
    ylabel('y'); xlabel('iter (k)');
    subplot(2,2,3);
    stem(history.z);
    ylabel('z'); xlabel('iter (k)');
    subplot(2,2,4);
    stem(history.objval)
    ylabel('(x^2+y^2+z^2)/2'); xlabel('iter (k)');

    K = length(history.objval);

    h = figure;
    plot(1:K, history.objval, 'k',...
```

```
            'MarkerSize', 10, 'LineWidth', 2);
        ylabel('(x^2+y^2+z^2)/2'); xlabel('iter (k)');

        g = figure;
        subplot(2,1,1);
        semilogy(1:K, max(1e-8, history.r_norm), 'k',...
            1:K, history.stol, 'k--',  'LineWidth', 2);
        ylabel('||r||_2');

        subplot(2,1,2);
        semilogy(1:K, max(1e-8, history.s_norm), 'k',...
            1:K, history.stol, 'k--', 'LineWidth', 2);
        ylabel('||s||_2'); xlabel('iter (k)');
    end
```

Matlab Code B.3: Variant of equalized method for counterexample

```
clc;
close all;
clear all;

%% Global constants and defaults
A=[1;1;1];
B=[1;1;2];
C=[1;2;2];
t_start = tic;
QUIET = 0;
MAX_ITER = 1000;
beta = 1;
tau = 1.1;
gamma = 1;
STOL = 1e-4;
%% Data preprocessing
[m, n]=size(A);
ATA=(A')*A;
BTB=(B')*B;
CTC=(C')*C;
AT=A';
BT=B';
CT=C';
%% Solver
x_k=1;
y_k=1;
z_k=1;
lambda_k=zeros(m,1);
if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\t%10s\t%10s\t%10s\n',...
    'iter','r norm', 's norm', 'objective', 'x', 'y', 'z');
end

for i=1:MAX_ITER
    x_old=x_k;
    x_k=(1+(tau+1)*beta*ATA)\(AT*lambda_k-beta*AT*B*y_k-...
        beta*AT*C*z_k+beta*tau*ATA*x_k);
    y_k=(1+(tau+1)*beta*BTB)\(BT*lambda_k-beta*BT*A*x_old-...
        beta*BT*C*z_k+beta*tau*BTB*y_k);
    lambda_p=lambda_k-beta*(A*x_k+B*y_k+C*z_k);
    z_k=(1+beta*CTC)\(CT*lambda_k-beta*CT*A*x_k-beta*CT*B*y_k);
    lambda_k=lambda_k-beta*(A*x_k+B*y_k+C*z_k);

    % diagnostics, reporting, termination checks
    history.x(i)=x_k;
    history.y(i)=y_k;
```

```
    history.z(i)=z_k;
    history.objval(i)=(x_k^2+y_k^2+z_k^2)/2;
    history.r_norm(i)=norm(A*x_k+B*y_k+C*z_k);
    history.s_norm(i)=norm(lambda_k-lambda_p);
    history.stol(i)=STOL;
    if ~QUIET
        fprintf('%3d\t\t%10.4f\t%10.4f\t%10.4f\t%10.4f\t
        %10.4f\t%10.4f\n', i, history.r_norm(i),...
        history.s_norm(i), history.objval(i),...
        history.x(i),history.y(i),history.z(i));
    end

    if (history.r_norm(i) < STOL && history.s_norm(i) < STOL)
        break;
    end
end
if ~QUIET
    toc(t_start);
    subplot(2,2,1);
    stem(history.x);
    ylabel('x'); xlabel('iter (k)');
    subplot(2,2,2);
    stem(history.y);
    ylabel('y'); xlabel('iter (k)');
    subplot(2,2,3);
    stem(history.z);
    ylabel('z'); xlabel('iter (k)');
    subplot(2,2,4);
    stem(history.objval)
    ylabel('(x^2+y^2+z^2)/2'); xlabel('iter (k)');

    K = length(history.objval);

    h = figure;
    plot(1:K, history.objval, 'k',...
        'MarkerSize', 10, 'LineWidth', 2);
    ylabel('(x^2+y^2+z^2)/2'); xlabel('iter (k)');

    g = figure;
    subplot(2,1,1);
    semilogy(1:K, max(1e-8, history.r_norm), 'k',...
        1:K,history.stol, 'k--', 'LineWidth', 2);
    ylabel('||r||_2');

    subplot(2,1,2);
    semilogy(1:K, max(1e-8, history.s_norm), 'k',...
        1:K, history.stol, 'k--', 'LineWidth', 2);
    ylabel('||s||_2'); xlabel('iter (k)');
end
```

B.2 Lasso with Elastic Net

Matlab Code B.4: Corrected method for Lasso with elastic net

```
clc;
close all;
clear all

randn('seed', 0);
rand('seed',0);
%% Global constants and defaults
```

```
m = 2000;          % number of examples
n = 2000;          % number of features
p = 100/n;         % sparsity density
x0 = sprandn(n,1,p);
K = randn(m,n);
K = K*spdiags(1./sqrt(sum(K.^2))',0,n,n); % normalize columns
b = K*x0 + sqrt(0.001)*randn(m,1);
lambda_1=1;
lambda_2=1;
A=[eye(n);eye(n)];
B=[-eye(n);zeros(n)];
C=[zeros(n);-eye(n)];
QUIET=0;
MAX_ITER=1000;
beta=1;
nu=0.9;
STOL=1e-3;
KTK=K'*K;
AT=A';
BT=B';
KT=K';
M=[eye(n),-(BT*B)\(BT*C);zeros(n),eye(n)];
%% Solver
t_start = tic;
x_k=zeros(n,1);
y_k=zeros(n,1);
z_k=zeros(n,1);
lambda_k=zeros(2*n,1);
if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\n','iter','r norm',...
            's norm' ,'objective');
end
for j=1:MAX_ITER
    v_old=[y_k;z_k];
    lambda_old=lambda_k;
    %Prediction:
    x_k=(2*KTK+beta*AT*A)\(2*KT*b+AT*lambda_k-...
        beta*AT*(B*y_k+C*z_k));
    y_k=((2*lambda_2+beta)*(eye(n))\(BT*lambda_k-...
        beta*BT*(A*x_k+C*z_k)));
    lambda_sub_2=lambda_k(n+1:2*n);
    for i=1:n
        z_k(i)= max( 0, x_k(i)-lambda_sub_2(i)/beta-...
        lambda_1/beta )-max( 0, -(x_k(i)-...
        lambda_sub_2(i)/beta ) - lambda_1/beta );
    end
    lambda_k=lambda_k-beta*(A*x_k+B*y_k+C*z_k);
    %Correction:
    v_k=[y_k;z_k];
    v_k=v_old-nu*M*(v_old-v_k);
    y_k=v_k(1:n);
    z_k=v_k(n+1:2*n);
    u_old=[v_old;lambda_old];
    u_k=[v_k;lambda_k];
    % diagnostics, reporting, termination checks
    history.objval(j)=sum((K*x_k - b).^2)+...
            lambda_2*sum(y_k.^2)+lambda_1*norm(z_k,1);
    history.r_norm(j)=norm(A*x_k+B*y_k+C*z_k);
    history.s_norm(j)=norm(u_k-u_old);
    history.stol(j)=STOL;
    if ~QUIET
      fprintf('%3d\t\t%10.4f\t%10.4f\t%10.4f\n', j,...
      history.r_norm(j),history.s_norm(j), history.objval(j));
    end
    if (history.r_norm(j) < history.stol(j) && ...
            history.s_norm(j) < history.stol(j))
```

```
            break;
        end
    end
    if ~QUIET
        toc(t_start);
        stem(x_k,'b');
        hold on
        stem(x0,'r');
        legend('x','x0');
        hold off
        ylabel('x'); xlabel('iter (k)');
        T = length(history.objval);
        h = figure;
        plot(1:T, history.objval , 'k', 'MarkerSize', 10,...
              'LineWidth', 2);
        ylabel('(norm(K*x-b))^2+lambda_1*(norm(x))^2+...
               lambda_2*norm(x,1)');
        xlabel('iter (k)');
        g = figure;
        subplot(2,1,1);
        semilogy(1:T, max(1e-8, history.r_norm), 'k', ...
            1:T, history.stol , 'k--', 'LineWidth', 2);
        ylabel('||r||_2');
        subplot(2,1,2);
        semilogy(1:T, max(1e-8, history.s_norm), 'k', ...
            1:T, history.stol , 'k--', 'LineWidth', 2);
        ylabel('||s||_2'); xlabel('iter (k)');
    end
```

Matlab Code B.5: Equalized method(Equalized method with relaxation factor) for Lasso with elastic net

```
clc;
close all;
clear all;

randn('seed', 0);
rand('seed',0);
%% Global constants and defaults
m = 2000;        % number of examples
n = 2000;        % number of features
p = 100/n;       % sparsity density
x0 = sprandn(n,1,p);
K = randn(m,n);
K = K*spdiags(1./sqrt(sum(K.^2))',0,n,n); % normalize columns
b = K*x0 + sqrt(0.001)*randn(m,1);
lambda_1=1;
lambda_2=1;
A=[eye(n);eye(n)];
B=[-eye(n);zeros(n)];
C=[zeros(n);-eye(n)];
QUIET     = 0;
MAX_ITER = 1000;
beta=1;
tau=1.1;
gamma=1.5;
STOL     = 1e-3;
KTK=K'*K;
AT=A';
BT=B';
KT=K';
%% Solver
t_start = tic;
x_k=zeros(n,1);
```

```matlab
y_k=zeros(n,1);
z_k=zeros(n,1);
lambda_k=zeros(2*n,1);
if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\n','iter','r norm',...
            's norm' ,'objective');
end
for j=1:MAX_ITER
    u_old=[y_k;z_k;lambda_k];
    x_k=(2*KTK+beta*AT*A)\(2*KT*b+AT*lambda_k-...
        beta*AT*(B*y_k+C*z_k));
    lambda_p=lambda_k-beta*(A*x_k+B*y_k+C*z_k);
    y_k=((2*lambda_2+(tau+1)*beta)*(eye(n)))\(BT*lambda_k-...
        beta*BT*A*x_k-beta*BT*C*z_k+tau*beta*BT*B*y_k);
    lambdap_sub_2=lambda_p(n+1:2*n);
    for i=1:n
      z_k(i)=max(0,z_k(i)-lambdap_sub_2(i)/((tau+1)*beta)-...
      lambda_1/((tau+1)*beta) )-max( 0, -(z_k(i)-...
    lambdap_sub_2(i)/((tau+1)*beta))-lambda_1/((tau+1)*beta));
    end
    lambda_k=lambda_k-gamma*beta*(A*x_k+B*y_k+C*z_k);
    u_k=[y_k;z_k;lambda_k];
    % diagnostics, reporting, termination checks
    history.objval(j)=sum((K*x_k - b).^2)+...
           lambda_1*sum(y_k.^2)+lambda_2*norm(z_k,1);
    history.r_norm(j)=norm(A*x_k+B*y_k+C*z_k);
    history.s_norm(j)=norm(u_k-u_old);
    history.stol(j)=STOL;
    if ~QUIET
      fprintf('%3d\t\t%10.4f\t%10.4f\t%10.4f\n', j,...
      history.r_norm(j), history.s_norm(j), history.objval(j));
    end
    if (history.r_norm(j) < history.stol(j) && ...
            history.s_norm(j) < history.stol(j))
        break;
    end
end
if ~QUIET
    toc(t_start);
    stem(x_k,'b');
    hold on
    stem(x0,'r');
    legend('x','x0');
    hold off
    ylabel('x'); xlabel('iter (k)');
    T = length(history.objval);
    h = figure;
    plot(1:T, history.objval, 'k', 'MarkerSize', 10,...
          'LineWidth', 2);
    ylabel('(norm(K*x-b))^2+lambda_1*(norm(x))^2+...
          lambda_2*norm(x,1)');
    xlabel('iter (k)');
    g = figure;
    subplot(2,1,1);
    semilogy(1:T, max(1e-8, history.r_norm), 'k', ...
        1:T, history.stol, 'k--', 'LineWidth', 2);
    ylabel('||r||_2');
    subplot(2,1,2);
    semilogy(1:T, max(1e-8, history.s_norm), 'k', ...
        1:T, history.stol, 'k--', 'LineWidth', 2);
    ylabel('||s||_2'); xlabel('iter (k)');
end
```

Matlab Code B.6: Variant of equalized method(Variant of equalized method with relaxation factor) for Lasso with elastic net

```
clc;
close all;
clear all;

randn('seed', 0);
rand('seed',0);
%% Global constants and defaults
m = 2000;        % number of examples
n = 2000;        % number of features
p = 100/n;       % sparsity density
x0 = sprandn(n,1,p);
K = randn(m,n);
K = K*spdiags(1./sqrt(sum(K.^2))',0,n,n); % normalize columns
b = K*x0 + sqrt(0.001)*randn(m,1);
lambda_1=1;
lambda_2=1;
A=[eye(n);eye(n)];
B=[-eye(n);zeros(n)];
C=[zeros(n);-eye(n)];
QUIET    = 0;
MAX_ITER = 1000;
beta=1;
tau=1.1;
gamma=1.5;
STOL    = 1e-3;
KTK=K'*K;
AT=A';
BT=B';
KT=K';
%% Solver
t_start = tic;
x_k=zeros(n,1);
y_k=zeros(n,1);
z_k=zeros(n,1);
lambda_k=zeros(2*n,1);
if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\n','iter','r norm',...
            's norm','objective');
end
for j=1:MAX_ITER
    u_old=[y_k;z_k;lambda_k];
    x_old=x_k;
    x_k=(2*KTK+(tau+1)*beta*AT*A)\(2*KT*b+AT*lambda_k-...
        beta*AT*(B*y_k+C*z_k)+tau*beta*AT*A*x_k);
    y_k=((2*lambda_2+(tau+1)*beta)*(eye(n)))\(BT*lambda_k-...
        beta*BT*A*x_old-beta*BT*C*z_k+tau*beta*BT*B*y_k);
    lambda_sub_2=lambda_k(n+1:2*n);
    for i=1:n
        z_k(i)=max(0,x_k(i)-lambda_sub_2(i)/beta-...
               lambda_1/beta )-max(0,-(x_k(i)-...
               lambda_sub_2(i)/beta)-lambda_1/beta);
    end
    lambda_k=lambda_k-gamma*beta*(A*x_k+B*y_k+C*z_k);
    u_k=[y_k;z_k;lambda_k];
    % diagnostics, reporting, termination checks
    history.objval(j)  = sum((K*x_k - b).^2)+...
            lambda_1*sum(y_k.^2)+lambda_2*norm(z_k,1);
    history.r_norm(j)  = norm(A*x_k+B*y_k+C*z_k);
    history.s_norm(j)  = norm(u_k-u_old);
    history.stol(j)=STOL;
    if ~QUIET
        fprintf('%3d\t\t%10.4f\t%10.4f\t%10.4f\n', j,...
                history.r_norm(j), history.s_norm(j),...
```

```
                    history.objval(j));
    end
    if (history.r_norm(j) < history.stol(j) && ...
            history.s_norm(j) < history.stol(j))
        break;
    end
end
if ~QUIET
    toc(t_start);
    stem(x_k,'b');
    hold on
    stem(x0,'r');
    legend('x','x0');
    hold off
    ylabel('x'); xlabel('iter (k)');
    T = length(history.objval);
    h = figure;
    plot(1:T, history.objval, 'k', 'MarkerSize', 10,...
        'LineWidth', 2);
    ylabel('(norm(K*x-b))^2+lambda_1*(norm(x))^2+...
        lambda_2*norm(x,1)');
    xlabel('iter (k)');
    g = figure;
    subplot(2,1,1);
    semilogy(1:T, max(1e-8, history.r_norm), 'k', ...
        1:T, history.stol, 'k--', 'LineWidth', 2);
    ylabel('||r||_2');
    subplot(2,1,2);
    semilogy(1:T, max(1e-8, history.s_norm), 'k', ...
        1:T, history.stol, 'k--', 'LineWidth', 2);
    ylabel('||s||_2'); xlabel('iter (k)');
end
```

B.3 Lasso with Constraints

Matlab Code B.7: Corrected method for Lasso with constraints

```
clc;
close all;
clear all

randn('seed', 0);
rand('seed',0);
%% Global constants and defaults
m = 2000;        % number of examples
n = 2000;        % number of features
p = 100/n;       % sparsity density
x0 = sprandn(n,1,p);
K = randn(m,n);
K = K*spdiags(1./sqrt(sum(K.^2))',0,n,n); % normalize columns
b = K*x0 + sqrt(0.001)*randn(m,1);
lambda_1=1;
A=[eye(n);eye(n)];
B=[-eye(n);zeros(n)];
C=[zeros(n);-eye(n)];
QUIET=0;
MAX_ITER=1000;
beta=1;
nu=0.9;
STOL=1e-3;
KTK=K'*K;
```

```
AT=A';
BT=B';
KT=K';
M=[eye(n),-(BT*B)\(BT*C);zeros(n),eye(n)];
%% Solver
t_start = tic;
x_k=zeros(n,1);
y_k=zeros(n,1);
z_k=zeros(n,1);
lambda_k=zeros(2*n,1);
if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\n','iter','r norm',...
        's norm' ,'objective');
end
for j=1:MAX_ITER
    v_old=[y_k;z_k];
    lambda_old=lambda_k;
    %Prediction:
    x_k=(2*KTK+beta*AT*A)\(2*KT*b+AT*lambda_k-...
        beta*AT*(B*y_k+C*z_k));
    lambda_sub_1=lambda_k(1:n);
    for i=1:n
        y_k(i)=max(0,x_k(i)-lambda_sub_1(i)/beta-...
            lambda_1/beta)-max(0,-(x_k(i)-...
            lambda_sub_1(i)/beta)-lambda_1/beta);
    end
    lambda_sub_2=lambda_k(n+1:2*n);
    for i=1:n
        z_k(i)= max( 0, x_k(i)-lambda_sub_2(i)/beta );
    end
    lambda_k=lambda_k-beta*(A*x_k+B*y_k+C*z_k);
    %Correction:
    v_k=[y_k;z_k];
    v_k=v_old-nu*M*(v_old-v_k);
    y_k=v_k(1:n);
    z_k=v_k(n+1:2*n);
    u_old=[v_old;lambda_old];
    u_k=[v_k;lambda_k];
    % diagnostics, reporting, termination checks
    history.objval(j)  = sum((K*x_k - b).^2)+...
                         lambda_1*sum(y_k.^2);
    history.r_norm(j)  = norm(A*x_k+B*y_k+C*z_k);
    history.s_norm(j)  = norm(u_k-u_old);
    history.stol(j)=STOL;
    if ~QUIET
        fprintf('%3d\t\t%10.4f\t%10.4f\t%10.4f\n',...
            j, history.r_norm(j),...
            history.s_norm(j), history.objval(j));
    end
    if (history.r_norm(j) < history.stol(j) && ...
            history.s_norm(j) < history.stol(j))
        break;
    end
end
if ~QUIET
    toc(t_start);
    stem(x_k,'b');
    hold on
    stem(x0,'r');
    legend('x','x0');
    hold off
    ylabel('x'); xlabel('iter (k)');
    T = length(history.objval);
    h = figure;
    plot(1:T, history.objval, 'k', 'MarkerSize', 10,...
        'LineWidth', 2);
```

```
    ylabel('(norm(K*x-b))^2+lambda_1*(norm(x))^2');
    xlabel('iter (k)');
    g = figure;
    subplot(2,1,1);
    semilogy(1:T, max(1e-8, history.r_norm), 'k', ...
        1:T, history.stol, 'k--', 'LineWidth', 2);
    ylabel('||r||_2');
    subplot(2,1,2);
    semilogy(1:T, max(1e-8, history.s_norm), 'k', ...
        1:T, history.stol, 'k--', 'LineWidth', 2);
    ylabel('||s||_2'); xlabel('iter (k)');
end
```

Matlab Code B.8: Equalized method(Equalized method with relaxation factor) for Lasso with constraints

```
clc;
close all;
clear all;

randn('seed', 0);
rand('seed',0);
%% Global constants and defaults
m = 2000;        % number of examples
n = 2000;        % number of features
p = 100/n;       % sparsity density
x0 = sprandn(n,1,p);
K = randn(m,n);
K = K*spdiags(1./sqrt(sum(K.^2))',0,n,n); % normalize columns
b = K*x0 + sqrt(0.001)*randn(m,1);
lambda_1=1;
A=[eye(n);eye(n)];
B=[-eye(n);zeros(n)];
C=[zeros(n);-eye(n)];
QUIET    = 0;
MAX_ITER = 1000;
beta=1;
tau=1.1;
gamma=1;
STOL    = 1e-3;
KTK=K'*K;
AT=A';
BT=B';
KT=K';
%% Solver
t_start = tic;
x_k=zeros(n,1);
y_k=zeros(n,1);
z_k=zeros(n,1);
lambda_k=zeros(2*n,1);
if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\n','iter','r norm',...
        's norm' ,'objective');
end
for j=1:MAX_ITER
    u_old=[y_k;z_k;lambda_k];
    x_k=(2*KTK+beta*AT*A)\(2*KT*b+AT*lambda_k-...
        beta*AT*(B*y_k+C*z_k));
    lambda_p=lambda_k-beta*(A*x_k+B*y_k+C*z_k);
    lambdap_sub_1=lambda_p(1:n);
    for i=1:n
        y_k(i)=max(0,y_k(i)-lambdap_sub_1(i)/((tau+1)*beta)-...
            lambda_1/((tau+1)*beta))-max(0,-(z_k(i)-...
            lambdap_sub_1(i)/((tau+1)*beta))-...
```

```
                lambda_1/((tau+1)*beta));
    end
    lambdap_sub_2=lambda_p(n+1:2*n);
    for i=1:n
        z_k(i)=max(0,z_k(i)-lambdap_sub_2(i)/((tau+1)*beta));
    end
    lambda_k=lambda_k-gamma*beta*(A*x_k+B*y_k+C*z_k);
    u_k=[y_k;z_k;lambda_k];
    % diagnostics, reporting, termination checks
    history.objval(j)  = sum((K*x_k - b).^2)+...
                         lambda_1*sum(y_k.^2);
    history.r_norm(j)  = norm(A*x_k+B*y_k+C*z_k);
    history.s_norm(j)  = norm(u_k-u_old);
    history.stol(j)=STOL;
    if ~QUIET
        fprintf('%3d\t\t%10.4f\t%10.4f\t%10.4f\n',...
                j, history.r_norm(j),...
                history.s_norm(j), history.objval(j));
    end
    if (history.r_norm(j) < history.stol(j) && ...
            history.s_norm(j) < history.stol(j))
        break;
    end
end
if ~QUIET
    toc(t_start);
    stem(x_k,'b');
    hold on
    stem(x0,'r');
    legend('x','x0');
    hold off
    ylabel('x'); xlabel('iter (k)');
    T = length(history.objval);
    h = figure;
    plot(1:T, history.objval, 'k', 'MarkerSize', 10,...
        'LineWidth', 2);
    ylabel('(norm(K*x-b))^2+lambda_1*(norm(x))^2');
    xlabel('iter (k)');
    g = figure;
    subplot(2,1,1);
    semilogy(1:T, max(1e-8, history.r_norm), 'k', ...
        1:T, history.stol, 'k--', 'LineWidth', 2);
    ylabel('||r||_2');
    subplot(2,1,2);
    semilogy(1:T, max(1e-8, history.s_norm), 'k', ...
        1:T, history.stol, 'k--', 'LineWidth', 2);
    ylabel('||s||_2'); xlabel('iter (k)');
end
```

Matlab Code B.9: Variant of equalized method(Variant of equalized method with relaxation factor) for Lasso with constraints

```
clc;
close all;
clear all;

randn('seed', 0);
rand('seed',0);
%% Global constants and defaults
m = 2000;        % number of examples
n = 2000;        % number of features
p = 100/n;       % sparsity density
x0 = sprandn(n,1,p);
```

```
K = randn(m,n);
K = K*spdiags(1./sqrt(sum(K.^2))',0,n,n); % normalize columns
b = K*x0 + sqrt(0.001)*randn(m,1);
lambda_1=1;
A=[eye(n);eye(n)];
B=[-eye(n);zeros(n)];
C=[zeros(n);-eye(n)];
QUIET    = 0;
MAX_ITER = 1000;
beta=1;
tau=1.1;
gamma=1;
STOL    = 1e-3;
KTK=K'*K;
AT=A';
BT=B';
KT=K';
%% Solver
t_start = tic;
x_k=zeros(n,1);
y_k=zeros(n,1);
z_k=zeros(n,1);
lambda_k=zeros(2*n,1);
if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\n','iter','r norm',...
            's norm' ,'objective');
end
for j=1:MAX_ITER
    u_old=[y_k;z_k;lambda_k];
    lambdap=lambda_k-beta*(A*x_k+B*y_k+C*z_k);
    x_k=(2*KTK+(tau+1)*beta*AT*A)\(2*KT*b+AT*lambda_k-...
         beta*AT*(B*y_k+C*z_k)+tau*beta*AT*A*x_k);
    lambdap_sub_1=lambdap(1:n);
    for i=1:n
        y_k(i)=max(0,y_k(i)-lambdap_sub_1(i)/((tau+1)*beta)-...
               lambda_1/((tau+1)*beta))-max(0,-(y_k(i)-...
               lambdap_sub_1(i)/((tau+1)*beta))-...
               lambda_1/((tau+1)*beta));
    end
    lambda_sub_2=lambda_k(n+1:2*n);
    for i=1:n
        z_k(i)= max( 0, x_k(i)-lambda_sub_2(i)/beta );
    end
    lambda_k=lambda_k-gamma*beta*(A*x_k+B*y_k+C*z_k);
    u_k=[y_k;z_k;lambda_k];
    % diagnostics, reporting, termination checks
    history.objval(j)  = sum((K*x_k - b).^2)+...
                         lambda_1*sum(y_k.^2);
    history.r_norm(j)  = norm(A*x_k+B*y_k+C*z_k);
    history.s_norm(j)  = norm(u_k-u_old);
    history.stol(j)=STOL;
    if ~QUIET
        fprintf('%3d\t\t%10.4f\t%10.4f\t%10.4f\n',...
                j, history.r_norm(j),...
                history.s_norm(j), history.objval(j));
    end
    if (history.r_norm(j) < history.stol(j) && ...
            history.s_norm(j) < history.stol(j))
        break;
    end
end
if ~QUIET
    toc(t_start);
    stem(x_k,'b');
    hold on
    stem(x0,'r');
```

```
    legend('x','x0');
    hold off
    ylabel('x'); xlabel('iter (k)');
    T = length(history.objval);
    h = figure;
    plot(1:T, history.objval, 'k', 'MarkerSize', 10,...
          'LineWidth', 2);
    ylabel('(norm(K*x-b))^2+lambda_1*(norm(x))^2');
    xlabel('iter (k)');
    g = figure;
    subplot(2,1,1);
    semilogy(1:T, max(1e-8, history.r_norm), 'k', ...
        1:T, history.stol, 'k--', 'LineWidth', 2);
    ylabel('||r||_2');
    subplot(2,1,2);
    semilogy(1:T, max(1e-8, history.s_norm), 'k', ...
        1:T, history.stol, 'k--', 'LineWidth', 2);
    ylabel('||s||_2'); xlabel('iter (k)');
end
```

Acknowledgements

This thesis is supervised by Prof. Bingsheng He, and without his earnest guidance and advice, I cannot finish my graduation project successfully. The optimization seminar held by him gives me great help, and is also the basis on my further study of optimization and applied mathematics.

I'd like to convey my gratitude to Prof. Zhihong Xia, who plays an important role in my decision to continue my education in mathematics. With the lead of him, mathematics department in SUSTech is becoming better.

I would also like to thank the wonderful committee of professors. The committee members Prof. Xuefeng Wang, Prof. Jingzhi Li, Prof. Anyue Chen, Prof. Shunxiang Ouyang, Prof. Linlin Su, Prof. Qin Li, Prof. Jun Chen, Prof. Xuejun Jiang, Prof Caiheng Li and Prof. Xianke Zhang who have contributed to exceptional teaching during my undergraduate life, which have helped me get through my education.

Finally, I'd like to say thanks to my family and my classmates for their encouragement and support.

Yao Li

May,2017