# 凸优化和单调变分不等式的收缩算法

# —— 统一框架与应用

## 前言目录与简要说明

南京大学数学系　　何炳生

hebma@nju.edu.cn



一个科学家最大的本领就在于化复杂为简单, 用简单的方法去解决复杂的问题。　——　冯康

✠　先贤名言, 铭记在心。纵然可以没有本领, 也不迷失价值标准　✠

爱美之心, 人皆有之。数学往往被人认为是枯燥的, 计算更是被人看作是繁琐的。领悟了数学之美的数学工作者, 他的职业生涯才可能是充满乐趣的。

✠　世上三百六十行, 应是同一道理　✠

数学之美, 不是纯数学的专利。为应用服务的最优化方法研究, 同样可以追求简单与统一。简单, 他人才会看懂使用; 统一, 自己才有美的享受。

✠　发现了其中的数学之美, 研究才变得满怀激情和欲罢不能　✠

♣ 对凸规划和单调变分不等式的收缩算法的研究，我们始终追求简明统一的原则。一个统一框架，二个孪生方向，三个基本不等式。不同方向，竟可用相同步长。漂亮的形式，令我沾沾自喜。但也常问自己，或以平庸为神奇？

♣ 统一框架，指导我们针对问题设计算法，也帮助我们简化算法的收敛性证明。理论有了保证，还要计算验证。尺有所短，寸有所长。自己动手编程，多些计算体验，才能给初学者提些有价值的参考意见。

♣ 投影收缩算法的定义在 Springer 出版社1975 年出版的 Blum 和 Oettli 的德文专著中就有了说明，作者均是苏黎世大学的博士，Oettli 生前曾是 Math. Programming 的编委。出于对历史的敬畏和对前人工作的尊重，根据算法具有的收缩特征，我们把所述方法都称为收缩方法。

♣ 从研究出发点开始，介绍收缩算法的基本原理。分若干篇章，对问题类型和主要方法作了梳理。谈问题，说算法，讲应用，给基本算例，也附简单程序。一条主线，一个模式。读懂一个算法，理解后面的篇章就不要再费多少力气。

**One algorithm framework should be flexible enough to solve many problems !**



## 动因与说明

这个系列讲义中的研究工作，主要包括以下四个方面：投影收缩算法、变分不等式框架下定制的PPA 算法、交替方向法以及多元分裂算法. 促使我整理这些讲义的动因，是这些工作分别得到了不同学科的一些著名学者引用.

一. 投影收缩算法：投影收缩算法方面的论文，得到包括 UC Berkeley 计算机系 Jordan 教授在内的学者引用，相关内容被写进他们的附录. Jordan 教授是美国科学院和工程院院士. 投影收缩算法也被中科院武汉岩土力学研究所的学者成功用来解决了困扰岩土工程界多年的问题.

二. 变分不等式框架下的松弛 PPA 算法：在变分不等式框架中构造算法的优越性，得到越来越多的学者认可. 我们 2012 年在 SIAM Journal Image Science 发表的文章，初稿就被一些欧洲的学者在他们最新的研究中引用. 例如，Pock and Chambolle 的文章中说到，用 He and Yuan 的 PPA 形式，极大地简化了收敛性分析 (which greatly simplified the convergence analysis). Chambolle 是在图像学领域有很大影响的学者.

三. 交替方向法：最优化中采用分裂算法，就像现实生活中大工程需要分包. 从招收第一批博士生开始，在收缩算法框架下研究交替方向法就是我们的一个主要课题. 新应用领域的发现，使交替方向法渐得广泛认可. 我们多年前发表的这类方法中的一个自调比准则，被 Stanford 大学电子工程系 Boyd 教授在 2010 年的一篇综述文章中称为一个简

单而有效的公式 (A simple scheme that often works well), 对我们的分析依据也作了简要介绍. Boyd 教授是美国工程院院士, 2006 年世界数学家大会邀请报告人.

四．多个可分离算子的分裂算法：交替方向法处理的是含两个可分离算子的问题. 对多于两个可分离算子的问题, 同样需要易于实现的分裂方法. 基于到鞍点越来越近的收缩思想, 我们给出了一些解决此类问题的方法. 这类工作, 已经被 UCLA 数学系教授 Osher 的课题组在降维问题上应用. 他们在文章中说到, The method proposed by He, Tao and Yuan is appropriate for this application. Osher 教授是美国科学院院士, 2010 年世界数学家大会一小时邀请报告人.

好方法需要有好的理论结果保证. 2005 年以来, Nemirovski 和 Nestrov 等最优化理论与方法领域的世界数学家大会邀请报告人对一阶算法收敛速率表现出的浓厚兴趣, 激发了我们对这个系列讲义中主要方法收敛速率的研究热情. 最后两讲, 我们用简单的方法和简短的篇幅, 分别论证了投影收缩算法和交替方向法都具有 $O(1/k)$ 的收敛速率. 换句话说, 对给定的 $\epsilon > 0$, 经过 $O(1/\epsilon)$ 次迭代, 就能得到遍历意义下的 $\epsilon$ 近似解.

**授人以鱼不如授人以渔。一个好的优化方法, 应该是容易被工程师们掌握, 让人用来自己解决问题的方法。**

✠ 为相关学科所用, 恰是我们从事优化方法研究的本源追求 ✠



# 凸优化和单调变分不等式的收缩算法 — 目录

## 第一部分: 单调变分不等式的求解方法


第一讲 变分不等式作为多种问题的统一表述模式 1

第二讲 三个基本不等式和变分不等式的投影收缩算法 21

第三讲 单调变分不等式收缩算法中的两对孪生方法 43


## 第二部分: 凸优化问题 $\{\min f(x)|Ax = b, x \in X\}$ 的求解方法


第四讲 为线性约束凸优化问题定制的 PPA 算法及其应用 65

第五讲 线性约束凸优化问题基于松弛 PPA 的收缩算法 87

第六讲 * 线性约束凸优化扩展问题的 PPA 和松弛 PPA 算法 109

第七讲 基于增广 Lagrange 乘子法的 PPA 收缩算法 131

**第三部分: 基于投影梯度的收缩算法**

第八讲 基于梯度投影的凸优化收缩算法和下降算法 149

第九讲 * 线性约束凸优化基于对偶上升的自适应方法 169

第十讲 * 线性约束单调变分不等式的自适应投影收缩算法 187

**第四部分: 凸优化问题** $\min\left\{\theta_1(x)+\theta_2(y)\,\middle|\,\begin{array}{c}Ax+By=b\\ x\in X, y\in Y\end{array}\right\}$ **的交替方向法**

第十一讲 结构型优化的交替方向收缩算法 209

第十二讲 线性化的交替方向收缩算法 227

第十三讲 定制 PPA 算法意义下的交替方向法 245

第十四讲 定制 PPA 算法意义的线性化交替方向法 263



**第五部分: 多个可分离算子凸优化问题带简单校正的分裂方法**

第十五讲 三个可分离算子凸优化的平行分裂 ALM 乘子法 279

第十六讲 三个可分离算子凸优化的略有改动的交替方向法 297

第十七讲 多个可分离算子凸优化带回代的交替方向收缩算法 321

第十八讲 * 多个可分离算子凸优化带回代的线性化交替方向法 337

**第六部分: 计算复杂性分析**

第十九讲 L-连续单调变分不等式投影收缩算法的收敛速率 353

第二十讲 交替方向法的计算复杂性和收敛速率 375

**可只读不带 $*$ 的篇章. 分解降低难度, 整合把握方向, 是该系列讲义的主要思想. 懂些变分不等式的基本概念, 对凸优化收缩算法的设计和收敛性分析很有帮助.**

# 凸优化和单调变分不等式的收缩算法

## 第一讲：变分不等式作为多种问题的统一表述模式

Variational inequality is a uniform approach for different problems

南京大学数学系　何炳生
hebma@nju.edu.cn



变分不等式是一种统一的问题表述模式. 管理科学与统计计算中存在大量凸优化问题. 信号处理, 图像恢复, 矩阵完整化, 机器学习等信息技术领域中也有许多问题可以归结为（或松弛成）一个凸优化问题. 凸优化的一阶必要性条件就是一个单调变分不等式. 在变分不等式的框架下研究凸优化的求解方法, 就像微积分中用导数求一元函数的极值, 常常会带来很大的方便.

除了通常的最优化问题以外, 互补问题是约束为正卦限的变分不等式. 经济活动中的空间价格平衡, 保护资源– 保障供给中的调控手段, 用经济手段解决交通疏导等问题, 都可以用变分不等式（或其特殊形式互补问题）来描述. 这一讲叙述以下一些常见问题与变分不等式的关系. 那些主要想从这个系列讲义学习结构型凸优化问题一阶算法的读者, 这一讲中只需要了解第一节.

- **凸优化与单调变分不等式**
- **商品流通、保护资源、保障供给中的变分不等式**
- **交通疏导中的变分不等式**
- **广义线性规划问题的线性变分不等式**
- **最短距离和问题的线性变分不等式**
- **极小化最大特征值之变分不等式**

# 1 凸优化与单调变分不等式

**连续优化方法中一些代表性的数学模型**

- 简单约束优化问题 $\min\ \{f(x) \mid x \in \Omega\}$.
- 线性约束优化问题 $\min\{\theta(x) \mid Ax = b,\ x \in \mathcal{X}\}$.
- 结构型优化问题 $\min\{\theta_1(x) + \theta_2(y) | Ax + By = b,\ x \in \mathcal{X},\ y \in \mathcal{Y}\}$.
- 非线性互补问题 $x \ge 0,\quad F(x) \ge 0,\quad x^T F(x) = 0$.
- 变分不等式 $x \in \Omega,\qquad (x' - x)^T F(x) \ge 0,\ \ \forall x' \in \Omega$.

我们只讲述求解凸优化和单调变分不等式的收缩算法. 从一般问题 (非线性互补问题和变分不等式) 的求解方法 (L2– L3)开始; 然后利用变分不等式刻画凸优化问题最优性条件, 讨论线性约束优化问题的求解方法 (L4– L7).

L8 讲述求解简单约束优化问题的只用导数信息的收缩算法. L9 讨论线性约束优化问题对偶变量的收缩算法. L10 是只用梯度的线性约束凸优化的求解. L11– L14 讨论结构型优化问题的收缩算法. L15– L18 讨论多个可分离算子的结构型优化问题的收缩算法. 最后, L19– L20 对主要算法的收敛速率做了论述.



## 1.1 可微凸优化等价于一个特殊的单调变分不等式

设 $\Omega$ 是 $\Re^n$ 中的非空闭凸集, 我们先来讨论可微凸优化问题

$$\min\{\theta(x) \mid x \in \Omega\} \tag{1.1}$$

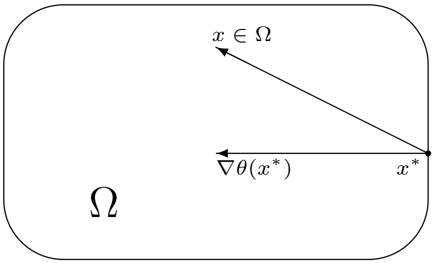
的最优性条件. 类似于瞎子爬山原理, 我们有

- 如果某一点 $x^*$ 是最优点, 它必须属于 $\Omega$,
- 并且从这点出发的所有可行方向都不是下降方向.

我们用 $\nabla\theta(x)$ 表示 $\theta(x)$ 的梯度, 并记

- $Sd(x) = \{s \in \Re^n \mid s^T \nabla\theta(x) < 0\}$, 为点 $x$ 处的下降方向集;
- $Sf(x) = \{s \in \Re^n \mid s = x' - x,\ x' \in \Omega\}$，为点 $x$ 处的可行方向集.

$$x^* \text{ 是最优解} \iff x^* \in \Omega \quad \text{且} \quad Sf(x^*) \cap Sd(x^*) = \emptyset$$




图示可微优化与变分不等式的关系

**上式可以写成：** $x^* \in \Omega, \quad (x - x^*)^T \nabla\theta(x^*) \geq 0, \quad \forall x \in \Omega.$

将 $\nabla\theta(x)$ 写成 $f(x)$, 可微凸优化问题 (1.1) 就归结为求

$$x \in \Omega, \quad (x' - x)^T f(x) \geq 0, \quad \forall x' \in \Omega. \tag{1.2}$$

以上形式就是一个变分不等式.



## 1.2 线性约束的凸优化问题

我们考虑一般的线性约束凸优化问题

$$\min\{\theta(x) \mid Ax = b, \; x \in \mathcal{X}\}, \tag{1.3}$$

其中 $\theta : \Re^n \to \Re$ 是凸函数, $\mathcal{X} \subset \Re^n$ 是闭凸集, $A \in \Re^{m\times n}$, $b \in \Re^m$.

设 $\lambda$ 是 Lagrange 乘子, 问题 (1.3) 的 Lagrange 函数

$$L(x, \lambda) = \theta(x) - \lambda^T(Ax - b)$$

定义在 $\mathcal{X} \times \Re^m$ 上. Lagrange 函数的鞍点 $(x^*, \lambda^*) \in \mathcal{X} \times \Re^m$ 满足

$$L_{\lambda\in\Re^m}(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L_{x\in\mathcal{X}}(x, \lambda^*)$$

为了简便, 我们假设 $\theta(x)$ 是可微的, 并且记 $\nabla\theta(x) = f(x)$, 鞍点问题等价于

$$\left\{ \begin{array}{lll} x^* \in \mathcal{X}, & (x - x^*)^T(f(x^*) - A^T\lambda^*) \geq 0, & \forall\, x \in \mathcal{X}, \\ \lambda^* \in \Re^m, & (\lambda - \lambda^*)^T(Ax^* - b) \geq 0, & \forall\, \lambda \in \Re^m. \end{array} \right. \tag{1.4}$$

上述最优性条件的第二部分也实际上就是 $Ax^* = b$. 如果记

$$u = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \qquad F(u) = \begin{pmatrix} f(x) - A^T\lambda \\ Ax - b \end{pmatrix} \quad \text{and} \quad \Omega = \mathcal{X} \times \Re^m,$$

那么, (1.4) 可以写成变分不等式

$$\boxed{u^* \in \Omega, \qquad (u - u^*)^T F(u^*) \geq 0, \quad \forall\, u \in \Omega.}$$

当 (1.3) 中的等式约束 $Ax = b$ 改成不等式约束 $Ax \geq b$ 时, 相应的变分不等式中只要将 $\Omega = \mathcal{X} \times \Re^m$ 改成 $\Omega = \mathcal{X} \times \Re^m_+$.

如果凸函数 $\theta(x)$ 是非光滑的, 记 $F_a(u) = \begin{pmatrix} -A^T\lambda \\ Ax - b \end{pmatrix}$, 那么, 求 Lagrange 函数的鞍点就等价于求混合变分不等式 (有时就简称为变分不等式)

$$\boxed{u^* \in \Omega, \quad \theta(x) - \theta(x^*) + (u - u^*)^T F_a(u^*) \geq 0, \quad \forall\, u \in \Omega}$$

的解 $u^*$. 记 $F_a(u) = Mu + q$, 则 $M$ 是斜对称矩阵, $F_a(u)$ 是仿射单调算子.



## 1.3 可分离结构的线性约束凸优化问题

可分离结构的线性约束凸优化问题是指

$$\boxed{\min\{\theta_1(x) + \theta_2(y) \mid Ax + By = b, \ x \in \mathcal{X}, \ y \in \mathcal{Y}\} \qquad (1.5)}$$

其中 $\theta_1(x)$ 和 $\theta_2(y)$ 是可微凸函数. 设 $\lambda$ 是 Lagrange 乘子, 上述问题的 Lagrange 函数

$$L(x, y, \lambda) = \theta_1(x) + \theta_2(y) - \lambda^T(Ax + By - b)$$

定义在 $\mathcal{X} \times \mathcal{Y} \times \Re^m$ 上, 设 $f(x) = \nabla\theta_1(x), \ g(y) = \nabla\theta_2(y)$. 这个可分离结构凸优化问题的一阶最优性条件是

$$(x^*, y^*, \lambda^*) \in \Omega, \quad \left\{ \begin{array}{l} (x - x^*)^T(f(x^*) - A^T\lambda^*) \geq 0, \\ (y - y^*)^T(g(y^*) - B^T\lambda^*) \geq 0, \qquad \forall\, (x, y, \lambda) \in \Omega, \\ (\lambda - \lambda^*)^T(Ax^* + By^* - b) \geq 0, \end{array} \right. \tag{1.6}$$

其中

$$\Omega = \mathcal{X} \times \mathcal{Y} \times \Re^m.$$

利用

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

最优性条件 (1.6) 可以写成

$$\boxed{w^* \in \Omega, \quad (w - w^*)^T F(w^*) \geq 0, \quad \forall\, w \in \Omega.}$$

当 $\theta_1(x), \theta_2(y)$ 是不可微凸函数时, 记 $u = (x, y)$ 为 $w$ 的部分分量, 并记 $\theta(u) = \theta_1(x) + \theta_2(y)$, 问题 (1.5) 等价于混合变分不等式

$$\boxed{w^* \in \Omega, \ \big(\theta(u) - \theta(u^*)\big) + (w - w^*)^T F_a(w^*) \geq 0, \ \forall\, w \in \Omega.}$$

我们在后面的章节中展示, 在变分不等式框架下研究凸优化求解方法, 无论在算法设计和收敛性证明方面, 会变得相当简单和方便.



## 1.4 非线性互补问题是一类特殊的变分不等式

凸优化问题 (1.1) 中, 如果 $\Omega = \Re^n_+$ (实 $n$-维空间中的非负卦限) 并且 $\theta(x)$ 是可微的, 根据 §1.1 中的分析, $x^*$ 是最优解的充要条件是

$$x^* \geq 0, \quad (x - x^*)^T \nabla\theta(x^*) \geq 0, \quad \forall\, x \geq 0.$$

设 $F$ 是 $\Re^n \to \Re^n$ 的一个算子. 非负卦限 $\Re^n_+$ 上变分不等式的形式是

$$\boxed{\textbf{VI}(\Re^n_+, F): \quad x \geq 0, \quad (x' - x)^T F(x) \geq 0, \quad \forall x' \geq 0.}$$

非线性互补问题是最优化理论与方法中一类很重要的问题. 它的数学形式是

$$\boxed{\textbf{(NCP)} \qquad x \geq 0, \quad F(x) \geq 0, \quad x^T F(x) = 0. \qquad (1.7)}$$

事实上, **NCP** 是 $\Omega = \Re^n_+$ 的一类变分不等式. 因此, 在 $\Omega = \Re^n_+$ 时, 可微凸优化问题 (1.1) 与一个互补问题等价.

**证明** 为帮助阅读理解, 我们给出**NCP** 与**VI**$(\Re^n_+, F)$ 等价的具体证明.

如果 $x$ 是 **NCP** 的解, 那么有 $x \geq 0$ 和 $F(x) \geq 0$.

对于任意的 $x' \geq 0$ 有 $(x')^T F(x) \geq 0$.

又因 $x^T F(x) = 0$, 得

$$\begin{aligned} (x' - x)^T F(x) \\ &= (x')^T F(x) \\ &\geq 0. \end{aligned}$$

所以 $x$ 是 **VI**$(R^n_+, F)$ 的一个解.

反过来, 如果 $x$ 是 **VI**$(R^n_+, F)$ 的一个解, 则 $x \geq 0$.

将 $x' = 0$ 和 $x' = 2x$ 代入 $(x' - x)^T F(x) \geq 0$, 我们得到 $\mp x^T F(x) \geq 0$. 因此 $x^T F(x) = 0$.

要证明 $x$ 是 **NCP** 的解, 只剩下 $F(x) \geq 0$ 需要证明, 对此采用反证法. 如果 $F(x)$ 的某个分量 $F_j(x) < 0$, 我们取 $x'$, 使得

$$x'_i = \begin{cases} x_i, & \text{if } i \neq j \\ x_j + 1, & \text{if } i = j \end{cases}$$

这样的 $x' \geq 0$. 但 $(x' - x)^T F(x) = F_j(x) < 0$, 这与 $x$ 是 **VI** 的解矛盾.



## 1.5 最小一模问题与等价的变分不等式

最小一模问题 (Least absolute deviations) 的数学形式是

$$\min \|Ax - b\|_1, \tag{1.8}$$

其中 $A \in \Re^{m\times n}, b \in \Re^m$. 与最小二乘相比, 它提供了一种更稳健的数据拟合模型. 与最小二乘不同, 它是一个非光滑凸优化问题. 这类问题被广泛应用于统计学和经济研究, 一些最新的文献都有提及. 例如:

- T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer, second edition, 2009. $\S 10.6$.
- J. M. Wooldridge. Introductory Econometrics: A Modern Approach. South Western College Publications, fourth edition, 2009. $\S 9.6$.

我们用 $e$ 表示每个分量都是 1 的 $m$-维向量. 容易验证

$$\|d\|_1 = \max\{y^T d \,|\, y \in B_\infty\}, \qquad B_\infty = \{y \in \Re^m \,|\, -e \leq y \leq e\}.$$

因此, 问题 (1.8) 就是一个形式为

$$\boxed{\min_{x\in\Re^n}\ \max_{y\in B_\infty}\ y^T(Ax-b).}$$

的 min-max 问题. 它的等价形式是下面的线性变分不等式:

$$\boxed{x^*\in\Re^n,\ y^*\in B_\infty,\quad \begin{cases}(x-x^*)^T(A^Ty^*)\ge 0, & \forall x\in\Re^n,\\ (y-y^*)^T(-Ax^*+b)\ge 0, & \forall y\in B_\infty.\end{cases}}$$

可以写成更简单紧凑的形式

$$\boxed{u^*\in\Omega,\quad (u-u^*)^T(Mu^*+q)\ge 0,\quad \forall u\in\Omega}$$

其中

$$u=\begin{pmatrix}x\\ y\end{pmatrix},\quad M=\begin{pmatrix}0 & A^T\\ -A & 0\end{pmatrix},\quad q=\begin{pmatrix}0\\ b\end{pmatrix},\quad \Omega=\Re^n\times B_\infty.$$



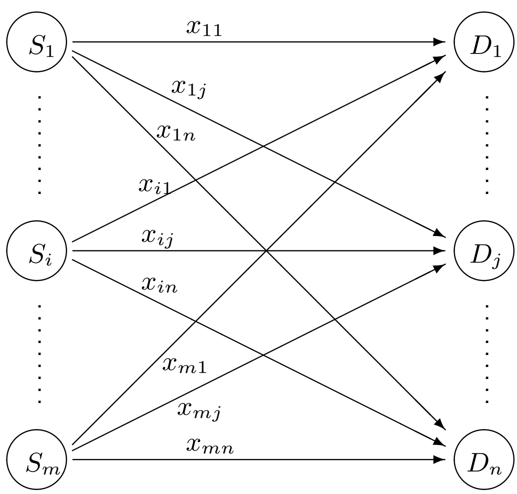
## 2 保护资源、保障供给中的互补问题

我们以下面的例子来说明一个经济平衡模型:

假设某种商品（例如煤）由 $m$ 个资源地生产和 $n$ 个需求地消费. 它由经营者们从资源地采购运到需求地销售. 经营者会根据贪婪原理找到他们的最优经营方案. 记

$s_i=\sum_{j=1}^n x_{ij}$

$d_j=\sum_{i=1}^m x_{ij}$



## 一些记号

$S_i$: 该种商品的第 $i$ 个资源地;

$D_j$: 该种商品的第 $j$ 个需求地;

$x_{ij}$: 从 $S_i$ 到 $D_j$ 的交易量;

$s_i$: 经营者们在资源地 $S_i$ 的采购总量, $s_i = \sum_{j=1}^{n} x_{ij}$;

$d_j$: 经营者在需求地 $D_j$ 的销售总量, $d_j = \sum_{i=1}^{m} x_{ij}$;

$h_i^s$: 经营者在资源地 $S_i$ 处的采购价, 是 $S_i$ 处采购量的函数;

$h_j^d$: 经营者在需求地 $D_j$ 处的销售价, 是 $D_j$ 处到货量的函数;

$t_{ij}$: 从 $S_i$ 到 $D_j$ 的交易费用 (包括运输费用等交易成本);

$y_i$: 政府为避免资源过度开采而在资源地 $S_i$ 向经营者征收的资源税;

$z_j$: 政府为保障供给而在需求地 $D_j$ 给经营者的经营补贴.



## 2.1 经营者追求利益最大化之互补问题

用下图表示第 $i$ 个资源地 $S_i$ 和第 $j$ 个需求地 $D_j$ 之间的采购— 销售关系.

采购价 $h_i^s$，资源税 $y_i$ 销售价 $h_j^d$，补贴 $z_j$

$$S_i \xrightarrow[\text{交易量 } x_{ij}]{\text{单位运输、劳务费用和必要的利润=交易费 } t_{ij}} D_j$$

如果$(h_i^s + y_i + t_{ij}) \geq (h_j^d + z_j)$, 没有人愿意做亏本买卖, 所以 $x_{ij} = 0$;

反之，根据贪婪原理，经营者会尽可能增大经营量 $x_{ij}$,（通常这会导致采购价的上涨和销售价的下降）直到

$$(h_i^s + y_i + t_{ij}) = (h_j^d + z_j).$$

用数学语言描述就是:

$$h_i^s + y_i + t_{ij} \begin{cases} \geq h_j^d + z_j, & \text{如果} x_{ij} = 0, \\ = h_j^d + z_j, & \text{如果} x_{ij} > 0\,. \end{cases}$$

- 对给定的资源税率 $y_1, y_2, \cdots, y_m$ 和补贴标准 $z_1, z_2, \cdots, z_n$ (政策),

  经营者们会根据“贪婪原理”找到他们的最优经营方案 $x_{ij}$ (对策).

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix},$$

如果记

$$F(X) = \begin{pmatrix} F_{11} & F_{12} & \dots & F_{1n} \\ F_{21} & F_{22} & \dots & F_{2n} \\ \vdots & \vdots & & \vdots \\ F_{m1} & F_{m2} & \dots & F_{mn} \end{pmatrix},$$



其中

$$F_{ij}(X) = \{h_i^s(s_i) + y_i + t_{ij}\} - \{h_j^d(d_j) + z_j\},$$

$$s_i = \sum_{j=1}^{n} x_{ij}, \qquad d_j = \sum_{i=1}^{m} x_{ij}.$$

那么, 由经营者的最优经营方案形成的非负 $m \times n$ 矩阵 $X$ 和 $F(X)$ 中, 同样下标的元素中最多只能有一个大于零. 换句话说, 经营者的最优经营方案是互补问题

$$\boxed{X \geq 0, \qquad F(X) \geq 0, \qquad \textbf{Trace}(X^T F(X)) = 0}$$

的解. 上式中的 **Trace**$(\cdot)$ 表示矩阵的“迹”—— 矩阵对角元的和.

- 我们能观测到的是问题解中相应的

$$\boxed{s_i(y,z) = \sum_{j=1}^{n} x_{ij}(y,z) \quad \textbf{and} \quad d_j(y,z) = \sum_{i=1}^{m} x_{ij}(y,z)}$$

它们是由政策向量 $(y, z)$ 决定的.

## 2.2 政府部门制定最优政策之互补问题

**政府部门的职责－保护资源和保障供给**

- 坚持可持续发展及环境保护，防止资源被过度开采，$s_i \le s_i^{\max}$;
- 保障基本供给，从而保证社会稳定和民生，$d_j \ge d_j^{\min}$.

**手段(政策)：**

- 在资源过度开采的产地征收资源税，$y_1, \cdots, y_m$；
- 供应紧张的需求地给经营者提供补贴，$z_1, \cdots, z_n$.

经营者们会根据政策给出对策. 政府的任务是给出‘最好’的政策.

**政府部门需要的最优政策是一个“黑箱”条件下的互补问题:**



- 保护了资源, 同时又让经济尽可能繁荣;

$$y \ge 0, \qquad s^{\max} - s(y,z) \ge 0, \qquad y^T(s^{\max} - s(y,z)) = 0,$$

- 保障了供给, 同时又尽可能节约财政支出.

$$z \ge 0, \qquad d(y,z) - d^{\min} \ge 0, \qquad z^T(d(y,z) - d^{\min}) = 0.$$

职能部门的互补问题的紧凑形式是：

$$u \ge 0, \qquad F(u) \ge 0, \qquad u^T F(u) = 0,$$

其中

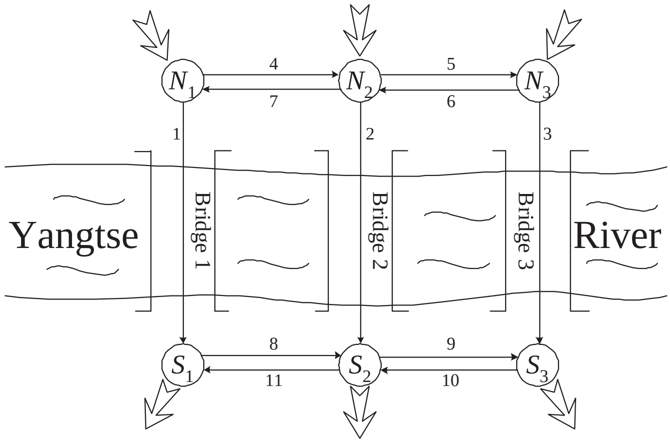
$$u = \begin{pmatrix} y \\ z \end{pmatrix}, \qquad F(u) = \begin{pmatrix} s^{\max} - s(u) \\ d(u) - d^{\min} \end{pmatrix},$$

$s(u)$ 和 $d(u)$ 是 $u$ 的函数.

**黑箱问题, 是指 $F(u)$ 是 $u$ 的函数, 有确定的关系, 但没有函数表达式.**

# 3 交通疏导问题之互补问题

设某跨江城市有三座长江大桥, 分别为 Bridge-1, Bridge-2 和 Bridge-3.




不失一般性, 我们可以将 $N_1, N_2, N_3$ 看作由北向南的车辆在江北的出发地, 把 $S_1, S_2, S_3$ 看作它们在江南的集散地.



我们只对管理部门的问题感兴趣, 假设只收过桥费. 管理部门想制定一个适当的收费标准合理控制桥上流量.

**驾驶员基于 Wardrop 原理的最优出行方案 — 最小费用路径**

对给定的大桥收费 $y = (y_1, y_2, y_3)$, 驾驶员会找到他们的最优出行方案.

**管理部门要求通过大桥合理收费控制桥上的流量**

- $0 \le y \in \Re^3$: 桥上的收费向量;
- $f(y) \in \Re^3$: 桥上的流量, 它是收费 $y$ 的函数;
- $0 < b \in \Re^3$: 管理部门希望控制的桥上的流量上界.

管理部门要求解的数学问题是

$$y \ge 0, \qquad F(y) = b - f(y) \ge 0, \qquad y^T F(y) = 0.$$

同样, 流量 $f(y)$ 确是收费 $y$ 的函数, 但没有表达式. 只能对给定的自变量, 观测相应的函数值, 而这种观测, 往往代价不菲. 我们从事投影收缩算法研究, 着眼点是要得到效率高一些的、只用函数值和少用函数值的方法.

# 4 广义线性规划及鞍点问题之变分不等式

## 4.1 广义线性规划及之变分不等式

线性规划的标准形式是 $\min\{c^T x \mid Ax = b, x \ge 0\}$. 其中 $A \in \Re^{m\times n}$, $b \in \Re^m$, $c \in \Re^n$. 在实际经济问题中, 向量 $b$ 一般表示需求量, $c$ 表示价格. 我们允许 $b$ 和 $c$ 都在一定范围之内变动, 考虑更一般的问题

$$\boxed{\min\{\max_{\eta\in C} \eta^T x \mid Ax \in B,\ x \in D\} \qquad (4.1)}$$

其中 $C, D \subset \Re^n$, $B \subset \Re^m$ 是闭凸集. 这样的问题我们称之为广义线性规划.
引进辅助变量 $y$ 和 Lagrange 乘子 $\lambda$, 得到广义线性规划 (4.1) 的 Lagrange 函数

$$L(x, y, \lambda, \eta) = \eta^T x - \lambda^T (Ax - y),$$

它定义在 $(D \times B) \times (\Re^m \times C)$ 上. 广义线性规划等价与一个 min-max 问题

$$L_{\lambda\in\Re^m, \eta\in C} L(x^*, y^*, \lambda, \eta) \le L(x^*, y^*, \lambda^*, \eta^*) \le L_{x\in D,\, y\in B}(x, y, \lambda^*, \eta^*).$$



设 $(x^*, y^*, \lambda^*, \eta^*)$ 是上述 min-max 问题的解, 则有

$$\left\{\begin{array}{lll} x^* \in D, & (x - x^*)^T(-A^T\lambda^* + \eta^*) \ge 0, & \forall\, x \in D \\ y^* \in B, & (y - y^*)^T(\lambda^*) \ge 0, & \forall\, y \in B \\ \lambda^* \in \Re^m, & (\lambda - \lambda^*)^T(Ax^* - y^*) \ge 0, & \forall\, \lambda \in \Re^m \\ \eta^* \in C, & (\eta - \eta^*)^T(-x^*) \ge 0, & \forall\, \eta \in C. \end{array}\right.$$

更紧凑的形式可以写成

$$\boxed{w^* \in \Omega, \quad (w - w^*)^T(Mw^* + q) \ge 0, \quad \forall w \in \Omega}$$

其中

$$w = \begin{pmatrix} x \\ y \\ \lambda \\ \eta \end{pmatrix}, \quad M = \begin{pmatrix} 0 & 0 & -A^T & I \\ 0 & 0 & I & 0 \\ A & -I & 0 & 0 \\ -I & 0 & 0 & 0 \end{pmatrix}, \quad q = 0$$

和

$$\Omega = D \times B \times \Re^m \times C.$$

一类常常被考虑的分裂可行问题 (Split Feasibility Problem)

$$\text{Find} \quad x \in D \quad \text{such that} \quad Ax \in B,$$

更是 (4.1) 问题中的一个特例. 将其转换成

$$\text{Find} \quad x \in D, \; y \in B, \quad \text{such that} \quad Ax - y = 0.$$

上述问题相当于一个目标函数为零的约束优化问题. 设 $\lambda$ 为线性约束 $Ax - y = 0$ 的 Lagrange 乘子, 问题就等价与以下的线性变分不等式:

$$\boxed{u^* \in \Omega, \quad (u - u^*)^T M u^* \ge 0, \quad \forall u \in \Omega}$$

其中

$$u = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, \quad M = \begin{pmatrix} 0 & 0 & -A^T \\ 0 & 0 & I \\ A & -I & 0 \end{pmatrix},$$

和

$$\Omega = D \times B \times \Re^m.$$



## 4.2 一般鞍点问题之变分不等式

用全变差极小处理图像去模糊 [1], 经离散化以后, 问题的数学模型是

$$\boxed{\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \; \Phi(x, y) := \theta_1(x) - y^T A x - \theta_2(y) \qquad (4.2)}$$

其中$\mathcal{X} \subset \Re^n$, $\mathcal{Y} \subset \Re^m$ 是闭凸集, $A \in \Re^{m \times n}$. $\theta_1(x) : \Re^n \to \Re$, $\theta_2(y) : \Re^m \to \Re$ 为凸函数. 如果 $(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$ 是问题 (4.2) 的解, 则有

$$\Phi_{y \in \mathcal{Y}}(x^*, y) \le \Phi(x^*, y^*) \le \Phi_{x \in \mathcal{X}}(x, y^*).$$

换句话说, $(x^*, y^*)$ 是 $\Phi(x, y)$ 在 $\mathcal{X} \times \mathcal{Y}$ 上的鞍点. 因此, 问题 (4.2) 能够转换成等价的变分不等式: 求 $(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$, 使得

$$\begin{pmatrix} x - x^* \\ y - y^* \end{pmatrix}^T \begin{pmatrix} f(x^*) - A^T y^* \\ g(y^*) + A x^* \end{pmatrix} \ge 0, \quad \forall (x, y) \in \mathcal{X} \times \mathcal{Y}, \qquad (4.3)$$

其中 $f(x) \in \partial\theta_1(x), g(y) \in \partial\theta_2(y)$. 用记号

$$u = \begin{pmatrix} x \\ y \end{pmatrix}, \quad F(u) = \begin{pmatrix} f(x) - A^T y \\ Ax + g(y) \end{pmatrix} \quad \text{and} \quad \Omega = \mathcal{X} \times \mathcal{Y}.$$

问题 (4.3) 就是变分不等式

$$u^* \in \Omega, \qquad (u - u^*)^T F(u^*) \geq 0, \quad \forall\, u \in \Omega. \tag{4.4}$$

由于 $\theta_1(x)$ 和 $\theta_2(y)$ 是凸函数, 容易验证, 这里的 $F(u)$ 是单调的. 关心这类问题求解方法的可以参阅第四讲与第五讲. 当 $\theta_1(x), \theta_2(y)$ 是非光滑凸函数时, 问题 (4.2) 也可以表述成混合变分不等式问题:

$$\theta(u) - \theta(u^*) + (u - u^*)^T M u^* \geq 0, \quad \forall\, u \in \Omega, \tag{4.5}$$

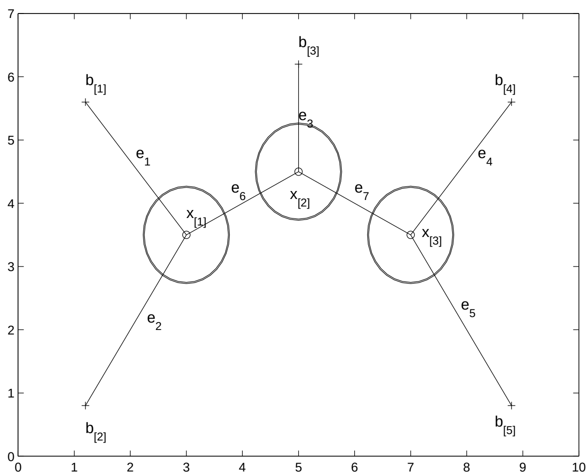
其中 $\theta(u) = \theta_1(x) + \theta_2(y)$, $M = \begin{pmatrix} 0 & -A^T \\ A & 0 \end{pmatrix}$ 是反对称矩阵. 换句话说, 混合变分不等式(4.5) 是单调的.



# 5 最短距离和问题之变分不等式

有些典型的非光滑凸优化问题可以化成结构相当简单的变分不等式. 这一节我们以最短距离和问题为例加以说明.

假设 $b_{[1]}, \ldots, b_{[5]}$ 是确定的村镇. 现在要用一个如右图的网络把它们连接起来. $x_{[1]}, x_{[2]}, x_{[3]}$ 是待选的连接点, 它们必须分别在划定的范围 $X_1, X_2, X_3$ 内. 如何确定 $x_{[1]}, x_{[2]}, x_{[3]}$ 的位置, 使得网络线路长度最短.



我们在 $p$-模意义下求上述网络的最短距离. 该问题的数学模型是

$$\min_{x_{[j]}\in X_j}\left\{\begin{array}{lll}\|x_{[1]}-b_{[1]}\|_p & +\|x_{[1]}-b_{[2]}\|_p & +\|x_{[2]}-b_{[3]}\|_p\\ & +\|x_{[3]}-b_{[4]}\|_p & +\|x_{[3]}-b_{[5]}\|_p\\ & +\|x_{[1]}-x_{[2]}\|_p & +\|x_{[2]}-x_{[3]}\|_p\end{array}\right\}. \tag{5.1}$$

主要对 $p=1,2,\infty$ 感兴趣. 注意到这里是距离和问题, 不是距离的平方和问题. 此类问题是一个非光滑凸优化问题.

## 5.1 欧氏模下的最短距离和问题

**将问题转化为 min-max 问题.** 注意到对任意的 $d\in\Re^2$, 有

$$\|d\|_2=\max_{\xi\in B_2}\xi^Td, \tag{5.2}$$

其中

$$B_2=\{\xi\in\Re^2\,|\,\|\xi\|_2\le 1\}.$$



**利用 (5.2), 上述欧氏模意义下的最短距离和问题可以化为min-max 问题**

$$\min_{x_{[i]}\in X_i}\max_{z_{[j]}\in B_2}\left\{\begin{array}{l}z_{[1]}^T(x_{[1]}-b_{[1]})+z_{[2]}^T(x_{[1]}-b_{[2]})+\ z_{[3]}^T(x_{[2]}-b_{[3]})\\ \qquad\qquad\qquad +\ z_{[4]}^T(x_{[3]}-b_{[4]})+\ z_{[5]}^T(x_{[3]}-b_{[5]})\\ \qquad\qquad\qquad +\ z_{[6]}^T(x_{[1]}-x_{[2]})+\ z_{[7]}^T(x_{[2]}-x_{[3]})\end{array}\right\}$$

它的紧凑形式为

$$\boxed{\min_{x\in\mathcal{X}}\max_{z\in\mathcal{B}_2}\ z^T(Ax-b)}$$

其中

$$\mathcal{X}=X_1\times X_2\times X_3\,,\qquad \mathcal{B}_2=B_2\times B_2\times\cdots\times B_2.$$

$$x=\begin{pmatrix}x_{[1]}\\ x_{[2]}\\ x_{[3]}\end{pmatrix},\quad z=\begin{pmatrix}z_{[1]}\\ z_{[2]}\\ \vdots\\ z_{[7]}\end{pmatrix}. \tag{5.3}$$

分块矩阵 $A$ 和向量 $b$ 的结构分别是

$$A = \begin{pmatrix} I_2 & 0 & 0 \\ I_2 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_2 \\ 0 & 0 & I_2 \\ I_2 & -I_2 & 0 \\ 0 & I_2 & -I_2 \end{pmatrix}, \qquad b = \begin{pmatrix} b_{[1]} \\ b_{[2]} \\ b_{[3]} \\ b_{[4]} \\ b_{[5]} \\ 0 \\ 0 \end{pmatrix}.$$

在这个 min-max 问题中, 每个 $x_{[i]}$ 对应一个点, 而每个 $z_{[j]}$ 则对应一条边.

设 $(x^*, z^*) \in \mathcal{X} \times \mathcal{B}_2$ 是 min-max 问题的解, 则对所有的 $x \in \mathcal{X}$ 和 $z \in \mathcal{B}$, 有

$$\boxed{z^T(Ax^* - b) \le {z^*}^T(Ax^* - b) \le {z^*}^T(Ax - b)}$$

它的等价形式是下面的线性变分不等式:



$$\boxed{x^* \in \mathcal{X},\ z^* \in \mathcal{B}_2,\quad \left\{ \begin{array}{ll} (x - x^*)^T(A^T z^*) \ge 0, & \forall x \in \mathcal{X}, \\ (z - z^*)^T(-Ax^* + b) \ge 0, & \forall z \in \mathcal{B}_2. \end{array} \right.}$$

可以写成更简单紧凑的形式

$$\boxed{u^* \in \Omega, \quad (u - u^*)^T(Mu^* + q) \ge 0, \quad \forall u \in \Omega}$$

其中有

$$u = \begin{pmatrix} x \\ z \end{pmatrix}, \quad M = \begin{pmatrix} 0 & A^T \\ -A & 0 \end{pmatrix}, \quad q = \begin{pmatrix} 0 \\ b \end{pmatrix}$$

和

$$\Omega = \mathcal{X} \times \mathcal{B}_2.$$

## 5.2 $l_1$-模下的最短距离和问题

由于对任意的 $d \in \Re^2$, 有

$$\|d\|_1 = \max_{\xi \in B_\infty} \xi^T d,$$

其中

$$B_\infty = \{\xi \in R^2 \mid \|\xi\|_\infty \le 1\}.$$

与欧氏模下的最短距离和问题一样, 问题 (5.1) 在 $l_1$-模意义下也可以表示成一个 $\min\text{-}\max$ 问题

$$\min_{x_{[i]} \in X_i} \max_{z_{[j]} \in B_\infty} \left\{ \begin{array}{rr} z_{[1]}^T(x_{[1]} - b_{[1]}) + z_{[2]}^T(x_{[1]} - b_{[2]}) + & z_{[3]}^T(x_{[2]} - b_{[3]}) \\ + \; z_{[4]}^T(x_{[3]} - b_{[4]}) + & z_{[5]}^T(x_{[3]} - b_{[5]}) \\ + \; z_{[6]}^T(x_{[1]} - x_{[2]}) + & z_{[7]}^T(x_{[2]} - x_{[3]}) \end{array} \right\}$$

它的紧凑形式为

$$\boxed{\min_{x \in \mathcal{X}} \max_{z \in \mathcal{B}_\infty} \quad z^T(Ax - b)}$$



与欧氏模下的距离和问题一样, 它等价于变分不等式

$$\boxed{u^* \in \Omega, \quad (u - u^*)^T(Mu^* + q) \ge 0, \quad \forall u \in \Omega}$$

矩阵 $M$ 和向量 $q$ 都不变, 所不同的只是这时集合

$$\Omega = \mathcal{X} \times \mathcal{B}_\infty, \qquad \mathcal{B}_\infty = B_\infty \times B_\infty \times \cdots \times B_\infty.$$

## 5.3 $l_\infty$-模下的最短距离和问题

由于对任意的 $d \in \Re^2$, 有

$$\|d\|_\infty = \max_{\xi \in B_1} \xi^T d,$$

其中

$$B_1 = \{\xi \in R^2 \mid \|\xi\|_1 \le 1\}.$$

与欧氏模下的最短距离和问题一样, 问题 (5.1) 在 $l_\infty$-模意义下也可以表示成

一个 min-max 问题

$$\min_{x_{[i]}\in X_i}\ \max_{z_{[j]}\in B_1}\left\{\begin{array}{r} z_{[1]}^T(x_{[1]}-b_{[1]})+z_{[2]}^T(x_{[1]}-b_{[2]})+\ z_{[3]}^T(x_{[2]}-b_{[3]}) \\ +\ z_{[4]}^T(x_{[3]}-b_{[4]})+\ z_{[5]}^T(x_{[3]}-b_{[5]}) \\ +\ z_{[6]}^T(x_{[1]}-x_{[2]})+\ z_{[7]}^T(x_{[2]}-x_{[3]}) \end{array}\right\}$$

并化成等价的变分不等式

$$\boxed{u^*\in\Omega,\quad (u-u^*)^T(Mu^*+q)\ge 0,\quad \forall u\in\Omega}$$

矩阵 $M$ 和向量 $q$ 都不变, 所不同的只是这时集合

$$\Omega=\mathcal{X}\times\mathcal{B}_1,\qquad \mathcal{B}_1=B_1\times B_1\times\cdots\times B_1.$$

将处理欧氏模问题时的 $\mathcal{B}_2$ 换成了$\mathcal{B}_1$. 我们会在第三讲具体介绍这类最短距离和问题的求解方法.



# 6 极小化最大特征值之变分不等式

设 $A_i, i=1,\ldots,m$ 和 $B$ 是给定的 $n\times n$ 对称矩阵, $c\in\Re^m$. 考虑问题

$$\boxed{\min_{x\in\mathcal{X}}\{\max\lambda_{\max}[\mathcal{A}(x)+B]+c^Tx\}\qquad (6.1)}$$

其中 $\mathcal{X}\subset\Re^m$,

$$\mathcal{A}(x)=\textstyle\sum_{i=1}^m x_iA_i. \tag{6.2}$$

这是经典半定规划中的对偶问题 [9]. 对给定的 $n\times n$ 矩阵 $A=(a_{ij})$, 矩阵 $A$ 的对角元的和称为矩阵 $A$ 的迹, 记为 **Tr**$(A)$, 即有

$$\textbf{Tr}(A)=\textstyle\sum_{j=1}^n a_{jj}.$$

熟知, 矩阵的特征值的和等于矩阵的迹. 对由 (6.2) 定义的 $\mathcal{A}(x)$ 和给定的 $n\times n$ 矩阵 $Y$, 我们用

$$\mathcal{A}^*(Y)=\begin{pmatrix}\textbf{Tr}(A_1Y)\\ \vdots\\ \textbf{Tr}(A_mY)\end{pmatrix}$$

定义 $\mathcal{A}^*(Y) \in \Re^m$. 在一些应用问题中, 集合 $\mathcal{X}$ 往往是个单纯形, 即 $\mathcal{X} = \{x \in \Re^m \mid \sum_{i=1}^m = 1,\ x \geq 0\}$. 注意到相似变换不改变矩阵特征值这一事实, 可以得到

$$\lambda_{\max}(A) = \max\{\mathbf{Tr}(YA) \mid \mathbf{Tr}(Y) = 1,\ Y \succeq 0\}.$$

所以, 问题 (6.1) 等价于下面的 $\min - \max$ 问题:

$$\min_{x\in\mathcal{X}} \max_{Y\succeq 0} \{\sum_{i=1}^m \big(x_i \mathbf{Tr}(YA_i)\big) + \mathbf{Tr}(YB) + c^T x \mid, \mathbf{Tr}(Y) = 1\}. \tag{6.3}$$

设 $(x^*, Y^*)$ 是问题 (6.3) 的解, 那么有 $\lambda^* \in \Re$, 使得

$$\Phi_{y\in\mathcal{S}_+^n, \lambda\in\Re}(x^*, Y, \lambda) \leq \Phi(x^*, Y^*, \lambda^*) \leq \Phi_{x\in\mathcal{X}}(x, Y^*, \lambda^*),$$

其中

$$\Phi(x, Y, \lambda) = \sum_{i=1}^m \big(x_i \mathbf{Tr}(YA_i)\big) + \mathbf{Tr}(YB) + c^T x + \lambda(\mathbf{Tr}(Y) - 1),$$

是问题 (6.3) 的 Lagrange 函数. 问题就转化为求 函数 $\Phi(x, Y, \lambda)$ 在 $\mathcal{X} \times (\mathcal{S}_+^n \times \Re)$ 上的鞍点 $(x^*, Y^*, \lambda^*)$. 因此, 问题 (6.1) 可以转换成下面的变分



不等式: 求 $(x^*, Y^*, \lambda^*) \in \mathcal{X} \times \mathcal{S}_+^n \times \Re$, 使得

$$\left\{\begin{array}{lll} x^* \in \mathcal{X}, & (x - x^*)^T\{\mathcal{A}^*(Y^*) + c\} \geq 0, & \forall x \in \mathcal{X}, \\ Y^* \succeq 0, & \langle Y - Y^*, -\mathcal{A}(x^*) - \lambda^* I - B\rangle \geq 0, & \forall Y \succeq 0, \\ \lambda^* \in \Re, & (\lambda - \lambda^*)(\mathbf{Tr}(Y^*) - 1) \geq 0, & \forall \lambda \in \Re. \end{array}\right.$$

上述变分不等式的紧致形式就是

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

其中

$$u = \begin{pmatrix} x \\ Y \\ \lambda \end{pmatrix}, \quad F(u) = \begin{pmatrix} \mathcal{A}^*(Y) + c \\ -\mathcal{A}(x) - \lambda I - B \\ \mathbf{Tr}(Y) - 1 \end{pmatrix} \quad \text{and} \quad \Omega = \mathcal{X} \times \mathcal{S}_+^n \times \Re.$$

**变分不等式是描述数学模型的强有力的统一工具**

当然, 这并不是说, 上面提及的问题都要化成变分不等式去求解. 从后面的篇章能清楚看到, 变分不等式的框架有利于设计算法和分析算法的收敛性质.

## 岩土力学中的许多工程问题可以归结为不同类型的变分不等式

岩土工程中的力学问题, 归结成的微分方程, 离散化以后得到的非线性变分不等式, 往往具备我们所说的单调性条件 [15]. 中科院武汉岩土力学研究所的科技工作者, 用 [4, 5] 中的方法解决了长期困扰岩土力学界的一些问题 [10, 11].

## References


[1] A. Chambolle and T. Pock, A first-order primal-dual algorithms for convex problem with applications to imaging, J. Math. Imaging Vison, **40**, 120-145, 2011.

[2] P.T. Harker, J.S.Pang, Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications, *Mathematical Programming*, **48**, 161-220, 1990.

[3] B.S. He, A projection and contraction method for a class of linear complementarity problems and its application in convex quadratic programming, Applied Mathematics and Optimization, **25**, 247-262, 1992.

[4] B.S. He, A class of projection and contraction methods for monotone variational inequalities, Applied Mathematics and Optimization, **35**, 69-76, 1997.

[5] B.S He and L-Z Liao, Improvements of some projection methods for monotone nonlinear variational inequalities, Journal of Optimization Theory and Applications, **112**, 111-128, 2002





[6] B.S. He, X.M. Yuan and J.J.Z. Zhang, Comparison of two kinds of prediction-correction methods for monotone variational inequalities, Computational Optimization and Applications, **27**, 247-267, 2004

[7] B. S. He and M.-H. Xu, A general framework of contraction methods for monotone variational inequalities, Pacific J. Optimization, **4**, 195-212, 2008.

[8] B.S. He, W. Xu , H. Yang, and X.M. Yuan, Solving resource protection and supply guarantee problems in economic equilibrium, Netw. Spat. Econ., 11,. 127-138, 2011.

[9] L. Vandenberghe and S. Boyd, Semidefinite programming, SIAM Review, **38**, 49-95, 1995.

[10] Hong Zheng, Feng Liu and Xiuli Du, Complementarity problem arising from static growth of multiple cracks and MLS-based numerical manifold method, Computer Methods in Applied Mechanics and Engineering, 295 (2015) 150-171.

[11] Hong Zheng, Peng Zhang and Xiuli Du, Dual form of discontinuous deformation analysis, Computer Methods in Applied Mechanics and Engineering, 305 (2016) 196-216.

[12] 何炳生, 论求解单调变分不等式的一些投影收缩算法, 〖计算数学〗, **18**, 54-60, 1996.

[13] 何炳生，邵虎，徐明华, 大桥流量分配调控中的隐式互补问题. 中国科技论文在线优秀论文集, 第三辑, 1-10, 2006.

[14] 何炳生, 徐薇, 杨海, 袁晓明, 经济平衡中的一类保护资源何保障供给问题, 中国科技论文在线优秀论文集, 第三辑, 11—19, 2006.

[15] 郑宏, 葛修润, 节理岩体的变分不等式模型, 中国科学(B辑) 25 卷, 第9期，986–993，1995.

# 凸优化和单调变分不等式的收缩算法

## 第二讲：三个基本不等式和 变分不等式的投影收缩算法

Three basic inequalities and the projection and contraction methods for variational inequalities

南京大学数学系　　何炳生
hebma@nju.edu.cn

The context of this lecture is mainly based on the publicatios [4, 6]



## 1 投影与变分不等式的一些基本性质

设 $\Omega \subset \Re^n$ 是一个闭凸集, $F$ 是从 $\Re^n$ 到自身的一个算子, 我们讨论单调变分不等式问题

$$\text{VI}(\Omega, F) \qquad u^* \in \Omega, \quad (u-u^*)^T F(u^*) \geq 0, \quad \forall u \in \Omega. \tag{1.1}$$

一个变分不等式称为单调的, 是指 VI$(\Omega, F)$ 中的 $F$ 是从 $\Re^n$ (或 $\Omega$) 到 $\Re^n$ 的一个单调算子 (monotone operator), 即 $F$ 满足

$$(F(u)-F(v))^T(u-v) \geq 0, \quad \forall\, u,\, v \in \Re^n (\text{or } \Omega).$$

说 $F$ 是单调仿射算子, 指 $F(u) = Mu + q$, $q \in \Re^n$, 其中 $M$ 是半正定矩阵. 一个 $n \times n$ 矩阵 $M$ 是半正定的, 是指对任何的 $u \in \Re^n$ 都有

$$u^T M u \geq 0.$$

这里并不要求矩阵 $M$ 对称. 换句话说, 只要 $M^T + M$ 对称半正定.

特别是, 当 $M$ 是反对称矩阵, 即 $M^T = -M$ 时, 总有 $u^T M u \equiv 0$. 这时, 仿射算子 $F(u) = Mu + q$ 是单调的.

设 $f(x)$ 是可微凸函数, 则它的导算子是单调的, 即有

$$(y-x)^T(\nabla f(y) - \nabla f(x)) \geq 0, \quad \forall x, y \in \Omega. \tag{1.2}$$

Let $\Omega \subset \Re^n$ be a convex closed set and $f$ be a convex function on $\Omega$. Assume that $f$ is differentiable on a open set that contains $\Omega$. Then $f$ is convex if and only if

$$f(y) - f(x) \geq \nabla f(x)^T(y-x), \quad \forall x, y \in \Omega. \tag{1.3}$$

这个结论可以在 Fletcher 的经典著作中

- R. Fletcher, Practical Methods of Optimization, Second Edition, §9.4. pp. 214–215, John Wiley & Sons, 1987.

中找到. 在 (1.3) 式中交换 $x$ 和 $y$ 的位置, 有

$$f(x) - f(y) \geq \nabla f(y)^T(x-y), \quad \forall x, y \in \Omega. \tag{1.4}$$

将 (1.3) 和 (1.4) 相加, 就得到 (1.2). 可微凸函数导算子的单调性得证.



## 1.1 投影的基本性质

用 $P_\Omega(\cdot)$ 表示欧氏范数下在凸集 $\Omega$ 上的投影, 也就是说

$$P_\Omega(v) = \text{Argmin}\{\|u-v\| \mid u \in \Omega\}.$$

如果 $\Omega = \Re^n_+$ ($n$-维空间的非负卦限), 那么 $P_\Omega(v)$ 的每个分量为

$$(P_\Omega(v))_j = \begin{cases} v_j, & \text{if} \quad v_j \geq 0; \\ 0, & \text{otherwise}. \end{cases}$$

如果 $\Omega$ 是 $n$-维空间中以 $c$ 为球心半径为 $r$ 的球, 那么

$$P_\Omega(v) = \begin{cases} \frac{r(v-c)}{\|v-c\|} + c, & \text{if} \quad \|v-c\| \geq r; \\ v, & \text{otherwise}. \end{cases}$$

在 $l_\infty$ 和 $l_1$ 模意义下的“单位球”上投影如下图所示：

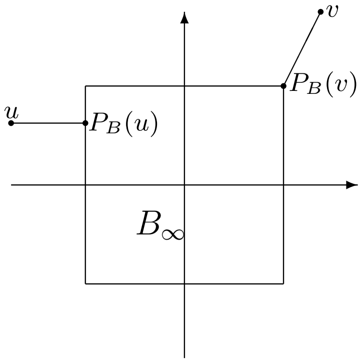



Fig. 1. Projection on $B_\infty$

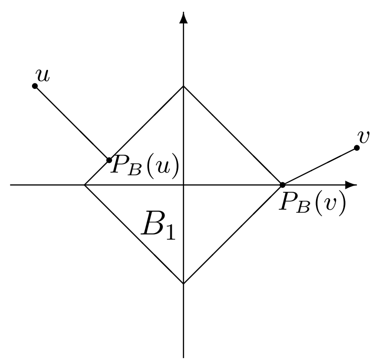



Fig. 2. Projection on $B_1$

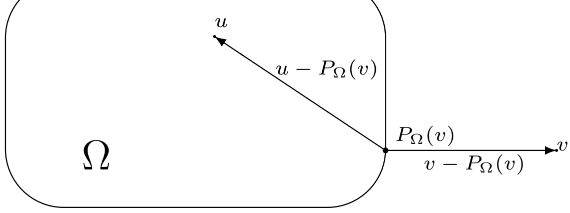
通篇, 我们记

$$\boxed{\tilde{u} = P_\Omega[u - F(u)].}$$



**Lemma 1.1** 设 $\Omega \subset R^n$ 是闭凸集, 则有

$$(v - P_\Omega(v))^T(u - P_\Omega(v)) \le 0 \qquad \forall v \in R^n, \forall u \in \Omega. \tag{1.5}$$




不等式(1.5) 的几何解释.

**证明.** 首先, 根据 $P_\Omega(v)$ 的定义, 有

$$\|v - P_\Omega(v)\| \le \|v - w\|, \quad \forall w \in \Omega.$$

注意到对任意的 $v \in \Re^n$, 都有 $P_\Omega(v) \in \Omega$, 由于 $\Omega \subset \Re^n$ 是闭凸集, 则对任意的 $u \in \Omega$ 和 $\theta \in (0,1)$, 都有

$$w := \theta u + (1-\theta)P_\Omega(v) = P_\Omega(v) + \theta(u - P_\Omega(v)) \in \Omega$$

和

$$\|v - P_\Omega(v)\|^2 \le \|v - P_\Omega(v) - \theta(u - P_\Omega(v))\|^2.$$

对上式展开, 就有对任意的 $u \in \Omega$ 和 $\theta \in (0,1)$,

$$[v - P_\Omega(v)]^T[u - P_\Omega(v)] \le \frac{\theta}{2}\|u - P_\Omega(v)\|^2.$$

令 $\theta \to 0_+$, 引理 (1.5) 得证. $\square$

在投影收缩算法的分析中, 不等式 (1.5) 是一个非常有用的基本工具. 我们因此而称之为 投影算子的工具不等式. 由(1.5), 容易证明下面的引理.

**Lemma 1.2** 设 $\Omega \subset R^n$ 是闭凸集, 则有

$$\|P_\Omega(v) - P_\Omega(u)\| \le \|v - u\|, \qquad \forall u, v \in R^n. \tag{1.6}$$

$$\|P_\Omega(v) - u\| \le \|v - u\|, \qquad \forall v \in R^n, u \in \Omega. \tag{1.7}$$

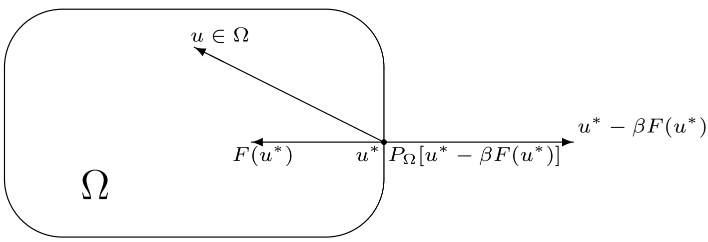
$$\|P_\Omega(v) - u\|^2 \le \|v - u\|^2 - \|v - P_\Omega(v)\|^2, \ \forall v \in R^n, u \in \Omega. \tag{1.8}$$



## 1.2 变分不等式等价的投影方程

设变分不等式 (1.1) 的解集 $\Omega^*$ 非空. 单调变分不等式的解集是凸的, 该结论可以在 [1] 的定理 2.3.5 中找到. 我们用 $u^*$ 表示一个确定的解点. 对任意的 $\beta > 0$, 变分不等式等价于投影方程

$$\boxed{u = P_\Omega[u - \beta F(u)].}$$




$u^*$ 是 VI($\Omega, F$) 的解等价于 $u^* = P_\Omega[u^* - \beta F(u^*)]$ 的几何解释

换言之, 求解变分不等式可以归结为求

$$e(u,\beta) := u - P_\Omega[u - \beta F(u)]$$

的一个零点 $u^*$, 后面我们会给出证明. 因此, $\|e(u,\beta)\|$ 可以看作一种误差的度量函数. 为了方便, 我们往往把 $e(u,1)$ 记成 $e(u)$.

**Theorem 1.1** 设 $\beta > 0$. $u^*$ 是 *VI*$(\Omega, F)$ 的解当且仅当 $e(u^*,\beta) = 0$.

**证明.** 先证必要性. 若 $u^*$ 是 VI$(\Omega, F)$ 的解, 则 $u^* \in \Omega$. 由于 $\Omega \subset R^n$ 是闭凸集, 利用 (1.5) 我们得到

$$\big(v - P_\Omega(v)\big)^T\big(u^* - P_\Omega(v)\big) \le 0, \qquad \forall v \in R^n.$$

上式中取 $v := u^* - \beta F(u^*)$, 则有 $\big(e(u^*,\beta) - \beta F(u^*)\big)^T e(u^*,\beta) \le 0$ ,即

$$\|e(u^*,\beta)\|^2 \le \beta e(u^*,\beta)^T F(u^*). \tag{1.9}$$

由于$P_\Omega[u^* - \beta F(u^*)] \in \Omega$, 而且 $u^*$ 是变分不等式的解, 根据 (1.1)可以得到

$$\{P_\Omega[u^* - \beta F(u^*)] - u^*\}^T F(u^*) \ge 0,$$



即,

$$e(u^*,\beta)^T F(u^*) \le 0. \tag{1.10}$$

由不等式 (1.9) 和 (1.10) 可得 $e(u^*,\beta) = 0$.

再证充分性. 取 $v = u^* - \beta F(u^*)$, 利用 (1.5) 和 $e(u^*,\beta)$ 的表达式, 有

$$\{e(u^*,\beta) - \beta F(u^*)\}^T\{u - P_\Omega[u^* - \beta F(u^*)]\} \le 0, \quad \forall u \in \Omega. \tag{1.11}$$

根据条件 $e(u^*,\beta) = 0$, 有 $u^* = P_\Omega(\cdot) \in \Omega$ 和 $P_\Omega[u^* - \beta F(u^*)] = u^*$. 代入不等式 (1.11), 可以得到

$$u^* \in \Omega, \qquad (u - u^*)^T F(u^*) \ge 0, \quad \forall u \in \Omega,$$

即 $u^*$ 是VI$(\Omega, F)$ 的解. 定理得证. $\square$

下面的定理说明 $\|e(u,\beta)\|$ 是 $\beta$ 的不减函数, 而 $\{\|e(u,\beta)\|/\beta\}$ 是 $\beta$ 的不增函数. 证明只用一元二次不等式的初等知识和工具不等式 (1.5), 它初见于 [13].

**Theorem 1.2** 对所有的 $u \in \Re^n$ 和 $\tilde\beta \ge \beta > 0$, 我们有

$$\|e(u,\tilde\beta)\| \ge \|e(u,\beta)\| \tag{1.12}$$

和

$$\frac{\|e(u,\tilde{\beta})\|}{\tilde{\beta}} \le \frac{\|e(u,\beta)\|}{\beta}. \tag{1.13}$$

**证明.** 设 $t = \|e(x,\tilde{\beta})\|/\|e(x,\beta)\|$, 定理的结论就相当于要证明 $1 \le t \le \tilde{\beta}/\beta$. 注意到它的等价表达式是 $t$ 的一元二次不等式

$$(t-1)(t-\tilde{\beta}/\beta) \le 0 \tag{1.14}$$

的解. 首先, 由工具不等式 (1.5), 我们有

$$(v - P_\Omega(v))^T(P_\Omega(v) - w) \ge 0, \quad \forall w \in \Omega. \tag{1.15}$$

在 (1.15) 中令 $w := P_\Omega[u - \tilde{\beta}F(u)]$ 和 $v := u - \beta F(u)$, 利用 $e(u,\beta)$ 的定义和

$$P_\Omega[u - \beta F(u)] - P_\Omega[u - \tilde{\beta}F(u)] = e(u,\tilde{\beta}) - e(u,\beta),$$

我们得到

$$\{e(u,\beta) - \beta F(u)\}^T\{e(u,\tilde{\beta}) - e(u,\beta)\} \ge 0. \tag{1.16}$$

用相应的方法(将上式中的 $\beta$ 和 $\tilde{\beta}$ 互换位置), 可得

$$\{e(u,\tilde{\beta}) - \tilde{\beta}F(u)\}^T\{e(u,\beta) - e(u,\tilde{\beta})\} \ge 0. \tag{1.17}$$



分别将不等式 (1.16) 和 (1.17) 乘上 $\tilde{\beta}$ 和 $\beta$, 然后再将它们相加, 我们得到

$$\{\tilde{\beta}e(u,\beta) - \beta e(u,\tilde{\beta})\}^T\{e(u,\tilde{\beta}) - e(u,\beta)\} \ge 0 \tag{1.18}$$

并有

$$\beta\|e(x,\tilde{\beta})\|^2 - (\beta+\tilde{\beta})e(x,\beta)^T e(x,\tilde{\beta}) + \tilde{\beta}\|e(x,\beta)\|^2 \le 0.$$

对上式采用 Cauchy-Schwarz 不等式, 就有

$$\beta\|e(x,\tilde{\beta})\|^2 - (\beta+\tilde{\beta})\|e(x,\beta)\| \cdot \|e(x,\tilde{\beta})\| + \tilde{\beta}\|e(x,\beta)\|^2 \le 0. \tag{1.19}$$

将 (1.19) 除 $\beta\|e(x,\beta)\|^2$, 并利用 $t$ 的定义便得

$$t^2 - \Big(1 + \frac{\tilde{\beta}}{\beta}\Big)t + \frac{\tilde{\beta}}{\beta} \le 0.$$

因此不等式 (1.14) 成立, 定理得证. $\square$

定理 1.2 说明, 若以 $\|e(u,\beta)\|$ 作为停机的误差度量, 常数 $\beta > 0$ 不宜过大, 也不宜过小. 一般要结合问题的物理意义考虑.

# 2 三个基本不等式和投影收缩算法

设 $u^*$ 是变分不等式 VI$(\Omega, F)$ 的解. 由于 $\tilde{u} = P_\Omega[u - \beta F(u)] \in \Omega$, 因此根据变分不等式的定义有第一个基本不等式

$$\boxed{\text{(FI1)} \quad (\tilde{u} - u^*)^T \beta F(u^*) \geq 0}$$

由于 $u^* \in \Omega$, $\tilde{u} = P_\Omega[u - \beta F(u)]$ 是 $u - \beta F(u)$ 在 $\Omega$ 上的投影. 根据投影的基本性质 (1.5), 我们有

$$\boxed{\text{(FI2)} \quad (\tilde{u} - u^*)^T \big([u - \beta F(u)] - \tilde{u}\big) \geq 0}$$

根据单调算子的性质, 有

$$\boxed{\text{(FI3)} \quad (\tilde{u} - u^*)^T (\beta F(\tilde{u}) - \beta F(u^*)) \geq 0}$$

一些投影收缩算法的寻查方向都是从这些基本不等式导出的.



## 投影收缩算法的基本框架

投影收缩算法顾名思义是基于投影的一种收缩算法. 对给定的 $\beta > 0$ 和当前点 $u^k$, 通过投影 $\tilde{u}^k = P_\Omega[u^k - \beta F(u^k)]$ 得到 $\tilde{u}^k$.

当前点 $u^k$ 是解点的充分必要条件是 $u^k = \tilde{u}^k$.

**误差度量函数** 一个非负函数 $\varphi(u^k, \tilde{u}^k)$ 称作变分不等式 VI$(\Omega, F)$ 的误差度量函数, 如果有 $\delta > 0$, 使得

$$\varphi(u^k, \tilde{u}^k) \geq \delta \|u^k - \tilde{u}^k\|^2, \quad \text{并且} \quad \varphi(u^k, \tilde{u}^k) = 0 \Leftrightarrow u^k = \tilde{u}^k. \tag{2.1a}$$

**有利方向** 一个向量 $d(u^k, \tilde{u}^k)$ 称为与误差度量函数 $\varphi(u^k, \tilde{u}^k)$ 相关的"有利方向"(或者称为距离函数的上升方向), 如果有

$$(u^k - u^*)^T d(u^k, \tilde{u}^k) \geq \varphi(u^k, \tilde{u}^k), \ \forall u^* \in \Omega^*. \tag{2.1b}$$

投影收缩算法的基本思想是根据变分不等式的性质, 构造一个方向 $d(u^k, \tilde{u}^k)$, 使它对一切 $u^* \in \Omega^*$ 都有 (2.1b) 成立. 我们把 $\varphi(u, \tilde{u})$ 称为"误差度量函数",

将 $d(u,\tilde{u})$ 称为“有利方向” (Profitable Direction). 尽管 $u^*$ 是我们要求的, 对任何一个给定的 $u^*$, $(u-u^*)$ 是距离函数 $\frac{1}{2}\|u-u^*\|^2$ 在 $u$ 处的梯度. (2.1b) 配上 (2.1a) 式表明 $-d(u^k,\tilde{u}^k)$ 是函数 $\frac{1}{2}\|u-u^*\|^2$ 在 $u^k$ 处的一个下降方向.

投影收缩算法 (Projection Contraction Method) 也可以看作是一种预估校正方法 (Prediction-Correction Method). 由投影得到 $\tilde{u}^k$ 是预估, 利用有利方向生成新的迭代点 $u^{k+1}$ 的过程为校正. 它们的英文简写, 都是 P-C Method.

我们要求投影收缩算法产生的迭代点到解点距离, 即 $\|u^k-u^*\|^2$ 严格单调下降. 这里先介绍 (2.1a) 和 (2.1b) 成立时, 由步长确定新迭代点的法则.

**一般收缩算法** 在条件 (2.1b) 满足时计算步长的迭代公式. 以

$$u^{k+1}(\alpha)=u^k-\alpha d(u^k,\tilde{u}^k) \tag{2.2}$$

产生依赖于步长 $\alpha$ 的新迭代点. 利用上式考察与 $\alpha$ 相关的距离平方缩短量,

$$\begin{aligned}\vartheta_k(\alpha) &= \|u^k-u^*\|^2-\|u^{k+1}(\alpha)-u^*\|^2\\ &= \|u^k-u^*\|^2-\|u^k-u^*-\alpha d(u^k,\tilde{u}^k)\|^2\\ &= 2\alpha(u^k-u^*)^Td(u^k,\tilde{u}^k)-\alpha^2\|d(u^k,\tilde{u}^k)\|^2.\end{aligned} \tag{2.3}$$



对任意给定的确定解点 $u^*$, (2.3) 表明 $\vartheta_k(\alpha)$ 是 $\alpha$ 的二次函数. 只是 $u^*$ 是未知的, 我们无法求它的极大. 不过, 利用 (2.1b) 有

$$\vartheta_k(\alpha)\ge 2\alpha\varphi(u^k,\tilde{u}^k)-\alpha^2\|d(u^k,\tilde{u}^k)\|^2. \tag{2.4}$$

将上式右端定义为 $q_k(\alpha)$, 我们得到 $\vartheta_k(\alpha)$ 的一个下界二次函数

$$q_k(\alpha)=2\alpha\varphi_k(u^k,\tilde{u}^k)-\alpha^2\|d(u^k,\tilde{u}^k)\|^2. \tag{2.5}$$

使二次函数 $q_k(\alpha)$ 达到极大的 $\alpha_k^*$ 是

$$\alpha_k^*=\varphi(u^k,\tilde{u}^k)/\|d(u^k,\tilde{u}^k)\|^2. \tag{2.6}$$

如果取

$$u^{k+1}=u^k-\alpha_k^*d(u^k,\tilde{u}^k). \tag{2.7}$$

由 (2.5) 和 (2.6), 得到

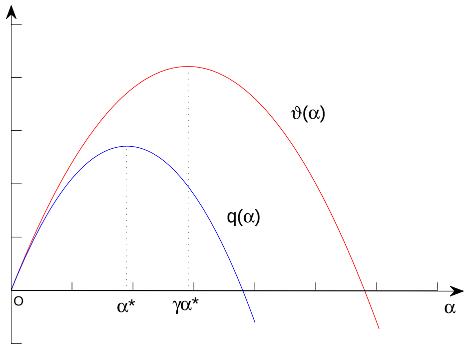
$$q_k(\alpha_k^*)=\alpha_k^*\varphi(u^k,\tilde{u}^k),$$

迭代公式 (2.7) 产生的序列 $\{u^k\}$ 不一定在 $\Omega$ 内, 却满足

$$\|u^{k+1}-u^*\|^2\le\|u^k-u^*\|^2-\alpha_k^*\varphi(u^k,\tilde{u}^k).$$

收缩算法的本意是想在每次迭代中极大化二次函数 $\vartheta_k(\alpha)$ (见 (2.3)), 由于它

含有未知的 $u^*$, 我们不得已才极大化它的下界函数 $q_k(\alpha)$.




取 $\gamma \in [1,2)$ 的示意图

因此, 在实际计算中, 我们一般取一个松弛因子 $\gamma \in [1,2)$, 令

$$u^{k+1} = u^k - \gamma\alpha_k^* d(u^k, \tilde{u}^k), \tag{2.8}$$

其中 $\gamma \in [1,2)$ 称为松弛因子. 取 $\gamma \in [1,2)$ 的理由可见示意图.



由 (2.5) 和 (2.6), 得到

$$\begin{aligned} q_k(\gamma\alpha_k^*) &= 2\gamma\alpha_k^*\varphi(u^k,\tilde{u}^k) - \gamma^2(\alpha_k^*)^2\|d(u^k,\tilde{u}^k)\|^2 \\ &= \gamma(2-\gamma)\alpha_k^*\varphi(u^k,\tilde{u}^k). \end{aligned}$$

公式 (2.8) 产生的序列 $\{u^k\}$ 满足

$$\|u^{k+1} - u^*\|^2 \le \|u^k - u^*\|^2 - \gamma(2-\gamma)\alpha_k^*\varphi(u^k,\tilde{u}^k). \tag{2.9}$$

利用 (2.6) 和 (2.8), 我们得到

$$\alpha_k^*\varphi(u^k,\tilde{u}^k) = \|\alpha_k^* d(u^k,\tilde{u}^k)\|^2 = \frac{1}{\gamma^2}\|u^k - u^{k+1}\|^2.$$

代入 (2.9), 我们得到与之等价的不等式

$$\|u^{k+1} - u^*\|^2 \le \|u^k - u^*\|^2 - \frac{2-\gamma}{\gamma}\|u^k - u^{k+1}\|^2.$$

**初等收缩算法.** 考虑将 (2.1a) 中的 $\varphi(u^k,\tilde{u}^k) \ge \delta\|u^k - \tilde{u}^k\|^2$ 改成条件

$$\varphi(u^k,\tilde{u}^k) \ge \frac{1}{2}\left(\|d(u^k,\tilde{u}^k)\|^2 + \tau\|u^k - \tilde{u}^k\|^2\right), \quad (\tau > 0). \tag{2.10}$$

我们将条件 (2.1b) 和 (2.10) 满足时, 用单位步长的迭代公式

$$u^{k+1} = u^k - d(u^k, \tilde{u}^k), \tag{2.11}$$

产生新迭代点的方法, 称为 **Primary Method** (初等方法). 简单计算可得

$$\begin{aligned} \|u^{k+1} - u^*\|^2 &= \|(u^k - u^*) - d(u^k, \tilde{u}^k)\|^2 \\ &= \|u^k - u^*\|^2 - 2(u^k - u^*)^T d(u^k, \tilde{u}^k) + \|d(u^k, \tilde{u}^k)\|^2 \\ &\leq \|u^k - u^*\|^2 - (2\varphi(u^k, \tilde{u}^k) - \|d(u^k, \tilde{u}^k)\|^2). \end{aligned}$$

因此, 由 (2.10) 产生的序列 $\{u^k\}$ 满足

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \tau\|u^k - \tilde{u}^k\|^2. \tag{2.12}$$

不等式 (2.9) 和 (2.12) 说明序列 $\{u^k\}$ 是有界的, 它们是算法收敛的关键式子. 利用 (2.1a) 和定理 1.1, 容易证明如下的定理.

**Theorem 2.1** 设 *VI*$(\Omega, F)$ 的解集 $\Omega^*$ 非空, 则投影收缩算法产生的序列 $\{u^k\}$ 收敛到 *VI*$(\Omega, F)$ 的某个解点 $u^* \in \Omega^*$.

需要说明的是, 在多数情况下, 我们主张采用计算步长确定新迭代点的一般收



缩算法. 虽然初等收缩算法不要额外计算步长, 但根据我们的计算经验, 一般收缩算法比初等收缩算法收敛快, 总的计算花费少.

♣ 可微凸优化问题 $\min\{f(x) \mid x \in \Omega\}$ 等价于变分不等式

$$x \in \Omega, \quad (x' - x)^T \nabla f(x) \geq 0, \quad \forall x' \in \Omega. \tag{2.13}$$

如果 $f(x)$ 二次可微, 它的 Hessian 矩阵 $\nabla^2 f(x)$ 是对称的. 特别当 $f(x)$ 是一个二次凸函数时, 它的 Hessian 矩阵 $H$ 是对称半正定的.

我们讲一般的非线性单调变分不等式VI$(\Omega, F)$, 只要求

$$(u - v)^T (F(u) - F(v)) \geq 0,$$

并不要求算子 $F$ 的 Jacobian 矩阵 $\nabla F(u)$ 对称. 当谈到单调线性变分不等式的时候, $F(u) = Mu + q$, 我们并不要求 $M$ 对称,

$$\text{只要求} \qquad M + M^T \qquad \text{半正定.}$$

可微凸优化问题是一类具有特殊性质的变分不等式 (2.13). 利用其 Hessian 矩阵的对称性, 能够设计一些效果更好的凸优化问题的收缩算法, 我们会在第八讲中讨论.

# 3 基于 FI1 + FI2 的 LVI 的投影收缩算法

这一节讨论的变分不等式 VI$(\Omega,F)$ 中, $F(u)=Mu+q$ 为仿射算子. 仍然记 $\tilde{u}=P_\Omega[u-\beta F(u)]$. 将第一和第二个基本不等式

$$\begin{cases} (\tilde{u}-u^*)^T\beta F(u^*)\ge 0. & (FI1) \\ (\tilde{u}-u^*)^T\big([u-\beta F(u)]-\tilde{u}\big)\ge 0. & (FI2) \end{cases}$$

相加, 对所有的 $u\in\Re^n$, 都有

$$\{(u-u^*)-(u-\tilde{u})\}^T\{(u-\tilde{u})-\beta M(u-u^*)\}\ge 0.$$

由上式直接得到

$$(u-u^*)^T(I+\beta M^T)(u-\tilde{u})\ge\|u-\tilde{u}\|^2+\beta(u-u^*)^TM(u-u^*).$$

由于 $M$ 半正定, 所以

$$(u-u^*)^T(I+\beta M^T)(u-\tilde{u})\ge\|u-\tilde{u}\|^2,\quad \forall u\in\Re^n. \tag{3.1}$$



令

$$\varphi(u,\tilde{u})=\|u-\tilde{u}\|^2 \tag{3.2}$$

和

$$d(u,\tilde{u})=(I+\beta M^T)(u-\tilde{u}). \tag{3.3}$$

由 (3.2) 定义的 $\varphi(u,\tilde{u})$ 和由 (3.3) 定义的 $d(u,\tilde{u})$ 满足条件 (2.1a) 和(2.1b).

其中的 $\delta=1$. 用一般收缩算法, 则由 (2.6) 计算步长 $\alpha_k^*$, 用迭代式 (2.8) 产生新的迭代点 $u^{k+1}$. 具体说来, 对给定的 $u^k$ 和常数 $\beta>0$, 令

$$\tilde{u}^k=P_\Omega[u^k-\beta(Mu^k+q)].$$

再用

$$u^{k+1}=u^k-\gamma\alpha_k^*(I+\beta M^T)(u^k-\tilde{u}^k)$$

产生新的迭代点, 其中

$$\alpha_k^*=\frac{\|u^k-\tilde{u}^k\|^2}{\|(I+\beta M^T)(u^k-\tilde{u}^k)\|^2}.$$

这样产生的序列 $\{u^k\}$ 满足

$$\|u^{k+1}-u^*\|^2 \le \|u^k-u^*\|^2-\gamma(2-\gamma)\alpha_k^*\|u^k-\tilde{u}^k\|^2.$$

在这个算法中, 参数 $\beta$ 的选取会严重影响收敛速度. 一般要用 "balance" 的思想, 动态地调整 $\beta_k$, 使得

$$\beta_k\|M^T(u^k-\tilde{u}^k)\| = \mathcal{O}(\|u^k-\tilde{u}^k\|).$$

♣ 如果所取的 $\beta$ 满足

$$\|(I+\beta M^T)(u^k-\tilde{u}^k)\|^2 \le (2-\tau)\|u^k-\tilde{u}^k\|^2, \quad \tau\in(0,1) \tag{3.4}$$

由 (3.2) 定义的 $\varphi(u,\tilde{u})$ 和由 (3.3) 定义的 $d(u,\tilde{u})$ 满足

$$2\varphi(u,\tilde{u}) \ge \|d(u,\tilde{u})\|^2+\tau\|u-\tilde{u}\|^2.$$

这说明“严格条件” (2.10) 成立, 就可以用步长为 $1$ 的初等收缩算法.

采用初等收缩算法的迭代公式 (2.11) 产生新的迭代点 $u^{k+1}$, 由 (2.12) 有

$$\|u^{k+1}-u^*\|^2 \le \|u^k-u^*\|^2-\tau\|u^k-\tilde{u}^k\|^2.$$

在下一讲, 我们将以与“最短距离和问题”为例, 讲述投影收缩算法的功效.



利用 FI1 和 FI2 相加得到求解线性变分不等式的更多方法, 可参见文献 [4] 和 [5]. 除了考虑欧氏模下的收缩算法, 我们也可以考虑正定矩阵 $G$-模下的收缩算法. 特别地, 若考虑 $G=(I+\beta M^T)(I+\beta M)$-模下 $\|u^k-u^*\|_G^2$ 的下降量. 则可采用迭代公式

$$u^{k+1}=u^k-\gamma(I+\beta M)^{-1}(u^k-\tilde{u}^k), \qquad \gamma\in(0,2). \tag{3.5}$$

根据上式就有

$$\begin{aligned}\|u^{k+1}-u^*\|_G^2 &= \|(u^k-u^*)-\gamma(I+\beta M)^{-1}(u^k-\tilde{u}^k)\|_G^2\\ &= \|u^k-u^*\|_G^2-2\gamma(u^k-u^*)^T(I+\beta M^T)(u^k-\tilde{u}^k)\\ &\quad +\gamma^2\|(I+\beta M)^{-1}(u^k-\tilde{u}^k)\|_G^2.\end{aligned}$$

利用 (3.1) 和 $G=(I+\beta M^T)(I+\beta M)$, 从上式得到

$$\begin{aligned}\|u^{k+1}-u^*\|_G^2 &\le \|u^k-u^*\|_G^2-2\gamma\|u^k-\tilde{u}^k\|^2+\gamma^2\|u^k-\tilde{u}^k\|^2\\ &= \|u^k-u^*\|_G^2-\gamma(2-\gamma)\|u^k-\tilde{u}^k\|^2.\end{aligned}$$

这个 $G$-模下的收缩算法, (3.5) 式不用计算步长, 但在整个迭代过程中要计算一次 $(I+\beta M)$ 的逆矩阵.

线性变分不等式中矩阵 $M$ 为对称半正定的情形

当矩阵 $M$ 为对称半正定的时候, 我们将其记为 $H$, (3.1) 就变成

$$(u-u^*)^T(I+\beta H)(u-\tilde{u}) \ge \|u-\tilde{u}\|^2, \quad \forall u \in \Re^n. \tag{3.6}$$

这时 $G = I + \beta H$ 是正定矩阵, 我们考虑 $G$-模下的收缩算法. 用

$$u^{k+1} = u^k - \gamma\alpha_k^*(u^k - \tilde{u}^k), \qquad \alpha_k^* = \frac{\|u^k-\tilde{u}^k\|^2}{\|u^k-\tilde{u}^k\|_G^2}, \qquad \gamma \in (0,2)$$

产生新的迭代点. 利用 (3.6) 就有

$$\begin{aligned}
\|u^{k+1}-u^*\|_G^2 &= \|(u^k-u^*) - \gamma\alpha_k^*(u^k-\tilde{u}^k)\|_G^2 \\
&= \|u^k-u^*\|_G^2 - 2\gamma\alpha_k^*(u^k-u^*)^T G(u^k-\tilde{u}^k) + \gamma^2(\alpha_k^*)^2\|u^k-\tilde{u}^k\|_G^2 \\
&\le \|u^k-u^*\|_G^2 - 2\gamma\alpha_k^*\|u^k-\tilde{u}^k\|^2 + \gamma^2(\alpha_k^*)^2\|u^k-\tilde{u}^k\|_G^2.
\end{aligned}$$

由于 $\alpha_k^*\|u^k-\tilde{u}^k\|_G^2 = \|u^k-\tilde{u}^k\|^2$, 根据上式, 迭代序列 $\{u^k\}$ 具有收缩性质

$$\|u^{k+1}-u^*\|_G^2 \le \|u^k-u^*\|_G^2 - \gamma(2-\gamma)\alpha_k^*\|u^k-\tilde{u}^k\|^2.$$



# 4 基于 FI1+FI2+FI3 的 NVI 的投影收缩算法

我们考虑求解一般的非线性单调变分不等式 (1.1) 的求解方法. 对给定的 $u$, 记 $\tilde{u} = P_\Omega[u - \beta F(u)]$. 将 (FI1), (FI2) 和 (FI3) 三个不等式

$$\begin{cases}
(\tilde{u}-u^*)^T\beta F(u^*) \ge 0 & (FI1) \\
(\tilde{u}-u^*)^T\big([u-\beta F(u)]-\tilde{u}\big) \ge 0 & (FI2) \\
(\tilde{u}-u^*)^T\big(\beta F(\tilde{u})-\beta F(u^*)\big) \ge 0 & (FI3)
\end{cases}$$

相加, 对所有的 $u \in \Re^n$, 都有

$$\boxed{(\tilde{u}-u^*)^T\{(u-\tilde{u}) - \beta[F(u)-F(\tilde{u})]\} \ge 0.}$$

我们定义

$$\varphi(u,\tilde{u}) = (u-\tilde{u})^T d(u,\tilde{u}), \tag{4.1}$$

和

$$d(u,\tilde{u}) = (u-\tilde{u}) - \beta\big(F(u)-F(\tilde{u})\big). \tag{4.2}$$

从而对一切的 $u \in \Re^n$, 都有

$$(u-u^*)^T d(u,\tilde{u}) \geq \varphi(u,\tilde{u}).$$

投影收缩算法中条件中的 (2.1b) 成立. 对于一个确定的 $\nu \in (0,1)$, 总可以采用**Armijo** 技术对算子进行调比, 使得

$$(u-\tilde{u})^T\big(\beta F(u)-\beta F(\tilde{u})\big) \leq \nu\|u-\tilde{u}\|^2 \tag{4.3}$$

式成立 (计算时往往同时使得 $\beta\|F(u)-F(\tilde{u})\| = \mathcal{O}(\|u-\tilde{u}\|)$. 根据 (4.2) 就有

$$\begin{aligned}\varphi(u,\tilde{u}) &= (u-\tilde{u})^T d(u,\tilde{u}) \\ &= \|u-\tilde{u}\|^2-(u-\tilde{u})^T\beta\big(F(u)-F(\tilde{u})\big) \\ &\geq (1-\nu)\|u-\tilde{u}\|^2.\end{aligned}$$

由 (4.1) 定义的 $\varphi(u,\tilde{u})$ 和由 (4.2) 定义的 $d(u,\tilde{u})$ 满足条件 (2.1a) 和(2.1b).

并且其中的 $\delta = 1-\nu$. 采用松弛迭代公式 (2.8) 求解单调变分不等式, 产生的序列 $\{u^k\}$ 满足

$$\|u^{k+1}-u^*\|^2 \leq \|u^k-u^*\|^2-\gamma(2-\gamma)\alpha_k^*(1-\nu)\|u^k-\tilde{u}^k\|^2.$$



♣ 特别地, 当

$$\beta\|F(u)-F(\tilde{u})\| \leq \nu\|u-\tilde{u}\| \tag{4.4}$$

时, 由 (4.2) 定义的 $d(u,\tilde{u})$ 和由 (4.1) 定义的 $\varphi(u,\tilde{u})$ 满足

$$\begin{aligned}&2\varphi(u,\tilde{u})-\|d(u,\tilde{u})\|^2 \\ &\quad= 2(u-\tilde{u})^T d(u,\tilde{u})-\|d(u,\tilde{u})\|^2 \\ &\quad= d(u,\tilde{u})^T\big\{2(u-\tilde{u})-d(u,\tilde{u})\big\} \\ &\quad= \big\{(u-\tilde{u})-\beta\big(F(u)-F(\tilde{u})\big)\big\}^T\big\{(u-\tilde{u})+\beta\big(F(u)-F(\tilde{u})\big)\big\} \\ &\quad= \|u-\tilde{u}\|^2-\beta^2\|F(u)-F(\tilde{u})\|^2 \;\geq\; (1-\nu^2)\|u-\tilde{u}\|^2.\end{aligned} \tag{4.5}$$

这说明条件 (2.10) 成立, 其中 $\tau = 1-\nu^2$.

采用 **Primary Method** 的迭代公式 (2.11), 根据 (2.12) 得到

$$\|u^{k+1}-u^*\|^2 \leq \|u^k-u^*\|^2-(1-\nu^2)\|u^k-\tilde{u}^k\|^2.$$

有关论文可见 [6, 11]. 对非对称的变分不等式, 即使 “严格条件” (4.4) 满足, 我们还是主张采用计算步长决定下一个迭代点的公式

$$u^{k+1} = u^k-\gamma\alpha_k^* d(u^k,\tilde{u}^k),$$

其中

$$\alpha_k^* = \frac{(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k)}{\|d(u^k, \tilde{u}^k)\|^2}.$$

在条件 (4.4) 满足时 (4.5) 左端严格大于零, 因而对所有的 $k$, 都有 $\alpha_k^* > 1/2$.

关于三个基本不等式的提法以及相关方法可见我的论文 [6] (初版是南京大学数学研究所 Preprint 94-11) 和 [14]. Sun Defeng 1992 年从南京大学硕士毕业后也独立发现用方向 (4.2) 构造投影收缩算法, 论文可见 [11]. 感谢 Sun 当年在论文 [11] 的 Footnote 和参考文献中都提到 [6].

**大道至简!** This approach has the benefit that three fundamental inequalities could be enough to construct contraction methods for many problems !

关于非线性单调变分不等式的求解, 已知的方法是外梯度方法 [9]. 这一讲我们只安排上述投影收缩算法与外梯度方法的比较.



# 5 外梯度方法

近几年来, 外梯度算法 (Extra-gradient methods) [9] 被北美一些名校 (宾习法尼亚大学, 2007, 多伦多大学, 2009, UC Berkeley, 2009, 哥伦比亚大学, 2009) 的博士们在语音识别、光纤网络、机器学习等研究中引用. 对如何提高外梯度算法算法收敛速度, 博士论文的作者建议进一步参考相关的文章 [8].

**外梯度算法** 外梯度算法, 实际上是对 PPA 算法采用预估一校正得来的.

Let us first briefly review the Proximal point Algorithm for VI$(\Omega, F)$ (1.1). PPA is an iterative method. For given $u^k$ and $r > 0$, the new iterate $u^{k+1}$ is the solution of the following variational inequality:

$$u^{k+1} \in \Omega, \quad (u - u^{k+1})^T \{F(u^{k+1}) + r(u^{k+1} - u^k)\} \geq 0, \ \forall\, u \in \Omega. \quad (5.1)$$

It is clear that $u^{k+1}$ is a solution of (1.1) if and only if $u^{k+1} = u^k$. In the case of $u^{k+1} \neq u^k$, by setting $u = u^*$ in (5.1), we obtain

$$(u^{k+1} - u^*)^T r(u^k - u^{k+1}) \geq (u^{k+1} - u^*)^T F(u^{k+1}).$$

Because $F$ is monotone, we have

$$(u^{k+1}-u^*)^TF(u^{k+1}) = (u^{k+1}-u^*)^TF(u^*) \geq 0$$

and consequently, we obtain

$$(u^{k+1}-u^*)^T(u^k-u^{k+1}) \geq 0,$$

and thus

$$(u^k-u^*)^T(u^k-u^{k+1}) \geq \|u^k-u^{k+1}\|^2.$$

By using the last inequality, we obtain

$$\|u^{k+1}-u^*\|^2 \leq \|u^k-u^*\|^2 - \|u^k-u^{k+1}\|^2. \tag{5.2}$$

**The sequence $\{u^k\}$ generated by PPA is Fejér monotone.**

The sequence $\{u^k\}$ generated by PPA has nice convergence property, however, the subproblem (5.1) is almost difficult as the original problem (1.1). Thus, the classical PPA is not widely used in the application.

By using $\beta = 1/r$ in (5.1), $u^{k+1}$ can be viewed as

$$u^{k+1} \in \Omega, \quad (u-u^{k+1})^T\{(u^{k+1}-u^k)+\beta F(u^{k+1})\} \geq 0, \ \forall\, u \in \Omega. \tag{5.3}$$



By using the equivalent representation of VI (see Theorem 1.1), it can be written as

$$u^{k+1} = P_\Omega\big[u^{k+1} - \{(u^{k+1}-u^k)+\beta F(u^{k+1})\}\big]$$

and thus

$$u^{k+1} = P_\Omega[u^k - \beta F(u^{k+1})]. \tag{5.4}$$

It is difficult to directly get the solution of (5.4). Replacing $u^{k+1}$ in the right hand side of (5.4) by $u^k$, we denote the output by

$$\tilde{u}^k = P_\Omega[u^k - \beta F(u^k)], \tag{5.5a}$$

and it called a predictor. Then, replacing $u^{k+1}$ in the right hand side of (5.4) by the predictor $\tilde{u}^k$, we obtain the (corrector) new iterate

$$u^{k+1} = P_\Omega[u^k - \beta F(\tilde{u}^k)] \tag{5.5b}$$

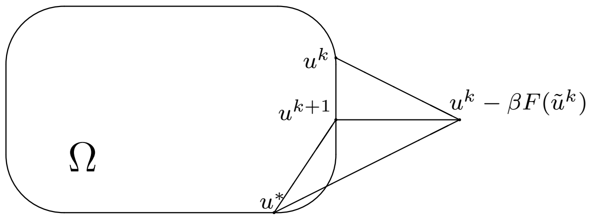
The method (5.5) is called the extra-gradient method (EG-method). Each iteration of the EG method includes two projections on $\Omega$. In the prediction step, the parameter $\beta$ should be chosen to satisfy the following condition:

$$\beta\|F(u^k)-F(\tilde{u}^k)\| \leq \nu\|u^k-\tilde{u}^k\|, \quad \nu \in (0,1). \tag{5.6}$$

**外梯度算法的收敛性分析** 外梯度算法收敛性的证明主要用投影的基本性质 (1.5) 和由它得到的 (1.8).

♣ 首先, 由于 $u^{k+1}$ 是 $u^k - \beta F(\tilde{u}^k)$ 到 $\Omega$ 上的投影, 根据投影性质 (1.8) 有

$$\|u^{k+1} - u^*\|^2 \le \|(u^k - \beta F(\tilde{u}^k)) - u^*\|^2 - \|(u^k - \beta F(\tilde{u}^k)) - u^{k+1}\|^2. \quad (5.7)$$




Geometric interpretation of inequality (5.7)

化简 (5.7) 式, 得到

$$\|u^{k+1} - u^*\|^2 \le \|u^k - u^*\|^2 - \|u^k - u^{k+1}\|^2 - 2(u^{k+1} - u^*)^T \beta F(\tilde{u}^k).$$



在上式右端加上 $(\tilde{u}^k - u^*)^T F(\tilde{u}^k) \ge (\tilde{u}^k - u^*)^T F(u^*) \ge 0$ 的这一项, 就有

$$\|u^{k+1} - u^*\|^2 \le \|u^k - u^*\|^2 - \|u^k - u^{k+1}\|^2 - 2(u^{k+1} - \tilde{u}^k)^T \beta F(\tilde{u}^k). \quad (5.8)$$

再将 (5.8) 式右端中的平方项 $\|u^k - u^{k+1}\|^2$ 写成

$$\|u^k - u^{k+1}\|^2 = \|u^k - \tilde{u}^k\|^2 + \|\tilde{u}^k - u^{k+1}\|^2 + 2(\tilde{u}^k - u^{k+1})^T (u^k - \tilde{u}^k),$$

并进一步整合, 就有

$$\begin{aligned} \|u^{k+1} - u^*\|^2 \quad &\le \quad \|u^k - u^*\|^2 - \|u^k - \tilde{u}^k\|^2 - \|\tilde{u}^k - u^{k+1}\|^2 \\ &\quad +2(u^{k+1} - \tilde{u}^k)^T \{u^k - \beta F(\tilde{u}^k) - \tilde{u}^k\}. \end{aligned} \quad (5.9)$$

♣ 接着, 我们对 $\tilde{u}^k$ 用投影的基本性质 (1.5). 由于 $\tilde{u}^k$ 是 $u^k - \beta F(u^k)$ 到 $\Omega$ 上的投影并且 $u^{k+1} \in \Omega$, 根据 (1.5), 有

$$2(\tilde{u}^k - u^{k+1})^T \{[u^k - \beta F(u^k)] - \tilde{u}^k\} \ge 0.$$

将上式加到 (5.9) 的右端, 得到

$$\begin{aligned} \|u^{k+1} - u^*\|^2 \quad &\le \quad \|u^k - u^*\|^2 - \|u^k - \tilde{u}^k\|^2 - \|\tilde{u}^k - u^{k+1}\|^2 \\ &\quad +2(u^{k+1} - \tilde{u}^k)^T \{\beta F(u^k) - \beta F(\tilde{u}^k)\}. \end{aligned} \quad (5.10)$$

♣ 对 (5.10) 式中右端包含的交叉项用 Cauchy–Schwarz 不等式,

$$\begin{aligned}&2(u^{k+1}-\tilde{u}^k)^T\{\beta F(u^k)-\beta F(\tilde{u}^k)\}\\ \leq\ \ & \|u^{k+1}-\tilde{u}^k\|^2+\beta^2\|F(u^k)-\beta F(\tilde{u}^k)\|^2,\end{aligned}$$

并代入 (5.10) 就得到

$$\|u^{k+1}-u^*\|^2 \leq \|u^k-u^*\|^2-\|u^k-\tilde{u}^k\|^2+\beta^2\|F(u^k)-F(\tilde{u}^k)\|^2. \quad (5.11)$$

**外梯度算法收缩性质** 根据 (5.11), 在条件 (5.6) 满足的情况下, 由外梯度法 (5.5) 生成的序列 $\{u^k\}$ 满足

$$\|u^{k+1}-u^*\|^2 \leq \|u^k-u^*\|^2-(1-\nu^2)\|u^k-\tilde{u}^k\|^2. \quad (5.12)$$

所以, 外梯度方法也是一个收缩算法. 这里给出的 (5.12) 是外梯度方法收敛的关键不等式. 上面提到的研究中使用外梯度算法, 已经用了我们在 [8] 中提出的一些优化策略. 证明这个结果需要用到的仅是投影算子的工具不等式 (1.5) 和 (1.8). 读者也可以参阅巨著 [1] 第二卷的 1115-1118 页关于外梯度算法的收敛性证明.



# 6 数值试验

这一节我们用求解单调非线性互补问题（变分不等式 VI$(\Omega,F)$ 中 $\Omega=\Re^n_+$ 的特殊形式）

$$u\geq 0, \qquad F(u)\geq 0, \qquad u^TF(u)=0,$$

来说明投影收缩算法比外梯度方法有明显优势. 在算例中我们取

$$F(u)=D(u)+Mu+q,$$

其中 $Mu+q$ 和 $D(u)$ 和别是 $F(u)$ 的线性和非线性部分. 生成线性部分 $Mu+q$ 采用类似于 [2][a] 的方式, 用下面的语句生成:

```
A=(rand(n,n)-0.5)*10;   B=(rand(n,n)-0.5)*10;    B=B-B';   M=A'*A+B;
q=(rand(n,1)-0.5)*1000;          或者           q=(rand(n,1)-1.0)*500;
```

非线性部分 $D(u)$ 的每个分量 $D_j(u)=d_j*\arctan(u_j)$, 其中 $d_j$ 是 $(0,1)$之间的随机数, 类似的取法可见 [12][b].

[a] In the paper by Harker and Pang [2], the matrix $M=A^TA+B+D$, where $A$ and $B$ are the same matrices as here, and $D$ is a diagonal matrix with uniformly distributed random variable $d_{jj}\in(0.0,0.3)$. In our test examples $d_{jj}\equiv 0$.

[b] In [12], the components of nonlinear mapping $D(u)$ are $D_j(u)=$ constat$*\arctan(u_j)$. Thus, $D_j(u)$ in our test example is more general.

在 [8] 中, 我们已经将外梯度方法精细化, 采用如下的程序:

Refined extra-gradient method:

Step 0. Set $\beta_0 = 1, \nu \in (0,1), u^0 \in \Omega$ and $k = 0$.

Step 1. $\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)]$,

$$r_k := \frac{\beta_k \|F(u^k) - F(\tilde{u}^k)\|}{\|u^k - \tilde{u}^k\|},$$

**while** $r_k > \nu, \quad \beta_k := \frac{2}{3}\beta_k * \min\{1, \frac{1}{r_k}\}$,

$\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)]$,

$$r_k := \frac{\beta_k \|F(u^k) - F(\tilde{u}^k)\|}{\|u^k - \tilde{u}^k\|},$$

**end(while)**

$u^{k+1} = P_\Omega[u^k - \beta_k F(\tilde{u}^k)]$,

**If** $r_k \le \mu$ then $\beta_k := \beta_k * 1.5$, **end(if)**

Step 2. $\beta_{k+1} = \beta_k$ and $k = k+1$, go to Step 1.

近几年人们在语音识别, 光纤网络, 机器学习等研究中使用外梯度算法时, 采用上述程序但略去 $\boxed{\textbf{If } r_k \le \mu \text{ then } \beta_k := \beta_k * 1.5 \ \textbf{end(if)}}$ 的做法.我们的计算实践说明, 如果略去这句话, 将大大增加迭代步数, 有时甚至导致计算失败.



相应的投影收缩算法是(较外梯度方法额外需要的计算量用小框标出):

Projection and Contraction Method:

Step 0. Set $\beta_0 = 1, \nu \in (0,1), u^0 \in \Omega$ and $k = 0$.

Step 1. $\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)]$,

$$r_k := \frac{\beta_k \|F(u^k) - F(\tilde{u}^k)\|}{\|u^k - \tilde{u}^k\|},$$

**while** $r_k > \nu, \quad \beta_k := \frac{2}{3}\beta_k * \min\{1, \frac{1}{r_k}\}$,

$\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)]$

$$r_k := \frac{\beta_k \|F(u^k) - F(\tilde{u}^k)\|}{\|u^k - \tilde{u}^k\|},$$

**end(while)**

$$\boxed{\begin{array}{l} d(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k) - \beta_k[F(u^k) - F(\tilde{u}^k)], \\ \alpha_k = \dfrac{(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k)}{\|d(u^k, \tilde{u}^k)\|^2}, \end{array}}$$

$u^{k+1} = u^k - \gamma\alpha_k d(u^k, \tilde{u}^k)$,

**If** $r_k \le \mu$ then $\beta_k := \beta_k * 1.5$, **end(if)**

Step 2. $\beta_{k+1} = \beta_k$ and $k = k+1$, go to Step 1.

外梯度算法和这里与之比较的投影收缩算法, 都可以看作是预估-校正方法. 它们用同样的公式

$$\tilde{u}^k = P_\Omega[u^k - \beta F(u^k)]$$

产生预估点 $\tilde{u}^k$. 为了进行效率比较, 我们都要求预估点满足 (见 (5.6))

$$\beta\|F(u^k) - F(\tilde{u}^k)\| \le \nu\|u^k - \tilde{u}^k\|, \quad \nu \in (0,1).$$

所不同的就是外梯度方法用 (见 (5.5b)) 校正公式

$$u^{k+1} = P_\Omega[u^k - \beta F(\tilde{u}^k)]$$

产生新的迭代点 $u^{k+1}$. 而投影收缩算法决定下一个迭代点的校正公式是

$$u^{k+1} = u^k - \gamma\alpha_k^* d(u^k, \tilde{u}^k),$$

其中方向由 (见 (4.2))

$$d(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k) - \beta\big(F(u^k) - F(\tilde{u}^k)\big),$$

给出. 步长则由

$$\alpha_k^* = (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k)/\|d(u^k, \tilde{u}^k)\|^2$$

确定. 投影收缩算法计算步长的额外工作量是不大的, 校正是不用投影的.



## 下面我们给出两种不同算法的 Matlab 程序

### The Matlab Code of the Refined extra-gradient Method

```
function REG(n,M,q,d,xstart,tol,pfq)                                 %(1)
fprintf('Extragragient Method by Korpelevich   n=%4d \n',n);          %(2)
x=xstart;  Fx= d.*atan(x) + M*x + q;  stopc=norm(x-max(x-Fx,0),inf);  %(3)
beta=1;        k=0;         l=0;        tic;                          %(4)
while (stopc>tol && k<=2000)                                          %(5)
  if mod(k,pfq)==0 fprintf(' k=%4d    epsm=%9.3e  \n',k,stopc); end;  %(6)
    x0=x;   Fx0=Fx;    k=k+1;                                         %(7)
    x=max(x0-Fx0*beta,0);        Fx=d.*atan(x) + M*x + q;   l=l+1;    %(8)
       dx=x0-x;    df=(Fx0-Fx)*beta;                                  %(9)
       r=norm(df)/norm(dx);                                           %(10)
       while r>0.9     beta=0.7*beta*min(1,1/r);    l=l+1;            %(11)
         x=max(x0-Fx0*beta,0);     Fx=d.*atan(x) + M*x + q;           %(12)
         dx=x0-x;    df=(Fx0-Fx)*beta;      r=norm(df)/norm(dx);      %(13)
       end;                                                           %(14)
       x=max(x0-Fx*beta,0);                                           %(15)
       Fx= d.*atan(x) + M*x + q;      l=l+1;                          %(16)
       ex=x-max(x-Fx,0);      stopc=norm(ex,inf);                     %(17)
       if r <0.4  beta=beta*1.5;  end;                                %(18)
end;  toc;   fprintf(' k=%4d   epsm=%9.3e     l=%4d  \n',k,stopc,l);  %%%%
```

将外梯度算法中第 (15) 行改成下面的第(15)-(16) 行, 就是投影收缩算法的程序.

## The Matlab Code of The Projection and Contraction Method

```
function PC_G(n,M,q,d,xstart,tol,pfq)                                  %(1)
fprintf('PC Method use Direction D1 with gamma a*  n=%4d \n',n);       %(2)
x=xstart;  Fx= d.*atan(x) + M*x + q;  stopc=norm(x-max(x-Fx,0),inf);   %(3)
beta=1;        k=0;        l=0;        tic;                            %(4)
while (stopc>tol && k<=2000)                                           %(5)
  if mod(k,pfq)==0 fprintf(' k=%4d    epsm=%9.3e  \n',k,stopc); end;   %(6)
    x0=x;    Fx0=Fx;    k=k+1;                                         %(7)
    x=max(x0-Fx0*beta,0);         Fx=d.*atan(x) + M*x + q;   l=l+1;    %(8)
       dx=x0-x;    df=(Fx0-Fx)*beta;                                   %(9)
       r=norm(df)/norm(dx);                                            %(10)
       while r>0.9     beta=0.7*beta*min(1,1/r);    l=l+1;             %(11)
         x=max(x0-Fx0*beta,0);     Fx=d.*atan(x) + M*x + q;            %(12)
         dx=x0-x;    df=(Fx0-Fx)*beta;     r=norm(df)/norm(dx);        %(13)
       end;                                                            %(14)
       dxf=dx-df;     r1=dx'*dxf;     r2=dxf'*dxf;     alpha=r1/r2;    %(15)
       x=x0- dxf*alpha*1.9;                                            %(16)
       Fx= d.*atan(x) + M*x + q;      l=l+1;                           %(17)
       ex=x-max(x-Fx,0);      stopc=norm(ex,inf);                      %(18)
       if r <0.4  beta=beta*1.5;  end;                                 %(19)
end;  toc;   fprintf(' k=%4d   epsm=%9.3e     l=%4d  \n',k,stopc,l);   %%%%
```



### NCP 的计算结果 1　Easy Problems $q \in (-500, 500)$

| | Extra-gradient Method | | | General PC-Method | | |
|---|---|---|---|---|---|---|
| $n =$ | No. It | No. F | CPU | No. It | No. F | CPU |
| 500 | 724 | 1485 | 0.26 | 468 | 977 | 0.17 |
| 1000 | 804 | 1650 | 2.85 | 514 | 1079 | 1.86 |
| 2000 | 776 | 1593 | 10.33 | 407 | 864 | 5.63 |

### NCP 的计算结果 2　Hard Problems $q \in (-500, 0)$

| | Extra-gradient Method | | | General PC-Method | | |
|---|---|---|---|---|---|---|
| $n =$ | No. It | No. F | CPU | No. It | No. F | CPU |
| 500 | 1453 | 2983 | 0.53 | 865 | 1824 | 0.33 |
| 1000 | 2026 | 4159 | 7.12 | 1199 | 2553 | 4.38 |
| 2000 | 1702 | 3494 | 22.45 | 1025 | 2177 | 14.00 |

The PC method converges faster than the refined extra-gradient method.

$$\frac{\text{It. No. of Projection and Contraction Method}}{\text{It. No. of The refined extra-gradient Method}} \approx 60\%.$$

♣ 程序在附件的 Codes-02 中：运行 demo.m 输入 $n$ 就可以, 其中也可以选择不同问题类型. REG.m 和 PC-G.m 分别是外梯度方法和投影收缩算法的子程序.

停机准则采用了绝对误差. 由于每步迭代的工作量是 $O(n^2)$ 的, 与问题规模比, 总的计算量是几个 $n^3$, 工作量是不大的. 投影收缩算法比外梯度算法每次迭代中要多计算一个步长, 工作量是 $O(n)$, 这在整个计算中比重是很小的.

需要指出, 在每次迭代中, 这里的收缩算法只做一次投影, 外梯度法要做两次投影. 在上面的算例中, 计算投影是轻而易举的. 而在机器学习的问题中, 有时所做的是到多面体上的投影, 投影占了总工作量的绝大部分. 对这类用外梯度方法求解的问题, 我们特别推荐改用本讲介绍的每次迭代只用一次投影的收缩算法. 它花费的总的计算时间会比精细的外梯度方法还节省 2/3.

## 采用Refined Extra-gradient methods 的一些博士论文

- Fei Sha, Large Margin Training of Acoustic Models for Speech Recognition, PhD Thesis, Computer and Information Science, University of Pennsylvania, 2007. 语音识别
- Yan Pan, A game theoretical approach to constrained OSNR optimization problems in optical network, PhD Thesis, Electrical & Computer Engineering. University of Toronto, 2009. 光纤网络
- Simon Lacoste-Julien, Discriminative Machine Learning with Structure, PhD Thesis, Computer Science. University of California, Berkeley, 2009. 机器学习
- A. G. Howard, Large Margin, Transformation Learning, PhD Thesis, Graduate School of Arts and Science. Columbia University, 2009. 机器学习



# References


[1] F. Facchinei and J.S. Pang, Finite-dimensional variational inequality and complementarity problems, Volume I and II, Springer Series in Operations Research, 2003.

[2] Harker, P.T. Harker and J.S. Pang *A damped-Newton method for the linear complementarity problem,* Lectures in Applied Mathematics **26**, 265–284, 1990.

[3] B.S. He, A projection and contraction method for a class of linear complementarity problems and its application in convex quadratic programming, Applied Mathematics and Optimization, **25**, 247–262, 1992.

[4] B.S. He, A new method for a class of linear variational inequalities, Math. Progr., **66**, 137-144, 1994.

[5] B.S. He, Solving a class of linear projection equations, Numerische Mathematik, **68**, 71-80, 1994.

[6] B.S. He, A class of projection and contraction methods for monotone variational inequalities, Applied Mathematics and Optimization, **35**, 69-76, 1997.

[7] B. S. He, Solving trust region problem in large scale optimization, J. Compu. Math. **18**, 1-12, 2000

[8] B.S He and L-Z Liao, Improvements of some projection methods for monotone nonlinear variational inequalities, Journal of Optimization Theory and Applications, **112**, 111-128, 2002

[9] G. M. Korpelevich. The extragradient method for finding saddle points and other problems, Ekonomika i Matematicheskie Metody, **12**, 747-756, 1976.

[10] D.F. Sun, A new step-size skill for solving a class of nonlinear projection equations, JCM, **13**, 357-368, 1995.

[11] D.F. Sun, A class of iterative methods for solving nonlinear projection equations, JOTA **91**, 123-140, 1996.

[12] K. Taji, M. Fukushima, and T. Ibaraki, A globally convergent Newton method for solving strongly monotone variational inequalities, Math. Progr. **58**, 369-383, 1993.

[13] T. Zhu and Z. G. Yu, A simple proof for some important properties of the projection mapping. Math. Inequal. Appl. **7**, 453–456, 2004.

[14] 何炳生, 论求解单调变分不等式的一些投影收缩算法, 〖计算数学〗, **18**, 54-60, 1996.

# 凸优化和单调变分不等式的收缩算法

## 第三讲：单调变分不等式投影收缩算法中的两对孪生方法

Two pairs of twin projection and contraction methods for monotone variational inequalities

南京大学数学系　何炳生

hebma@nju.edu.cn

The context of this lecture is based on the publication [8, 10, 13, 14]



我们研究单调变分不等式求解方法. 对任给的 $\beta > 0$, 变分不等式

$$\text{VI}(\Omega, F) \quad u^* \in \Omega, \quad (u - u^*)^T F(u^*) \geq 0, \quad \forall u \in \Omega \tag{0.1}$$

与 $F$ 经调比后的问题

$$\text{VI}(\Omega, F) \quad u^* \in \Omega, \quad (u - u^*)^T F(u^*) \geq 0, \quad \forall u \in \Omega$$

同解. 常用 $e(u, \beta) := u - P_\Omega[u - \beta F(u)]$ 的模 $\|e(u, \beta)\|$ 度量误差.

最近几年, 线性变分不等式的投影收缩算法 [6, 7] 被成功应用到机器人的运动规划和实时控制中 [5, 16]. 利用 [9, 10] 中求解非线性变分不等式的方法, 中科院武汉岩土力学研究所的学者成功解决一些岩土工程问题 [17, 18]. 本文分别介绍单调线性变分不等式和单调非线性变分不等式的两对孪生方法. 每对孪生方法都有投影提供一对孪生下降方向, 采用不同方式产生新迭代点的时候却采用相同的步长. 这些结果, 除了对提高算法效率有用, 从数学规律来讲, 也给了我们美的享受.

# 1 预备知识

**投影的基本性质**. 我们只列出几个这里需要的、显而易见的性质. 对这些性质证明感兴趣的读者可以参考作者主页上系列讲义的第二讲.

**Lemma 1.1** 设 $\Omega \subset R^n$ 是闭凸集, 则对任意的 $v \in \Re^n$, 有

$$(u - P_\Omega(v))^T (P_\Omega(v) - v) \geq 0, \quad \forall u \in \Omega. \tag{1.1}$$

**变分不等式所等价的投影方程**. 利用投影的概念, 求解变分不等式 (0.1) 可以归结为求

$$e(u,\beta) := u - P_\Omega[u - \beta F(u)] \qquad \text{的一个零点.}$$

求解单调变分不等式 (2) 的投影收缩算法是一种预测-校正方法. 由投影提供预测, 由校正实现收缩. 在投影收缩算法的 $k$-步迭代中, 对给定的当前点 $u^k$ 和 $\beta_k > 0$, 我们利用投影

$$\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)], \tag{1.2}$$



生成一个预测点 $\tilde{u}^k$. 根据变分不等式解的性质, $u^k = \tilde{u}^k$ 是 $u^k \in \Omega^*$ 的充分必要条件. 利用投影的基本性质 (1.1) 中, 令 $v = u^k - \beta_k F(u^k)$, 根据 (1.2), $\tilde{u}^k = P_\Omega(v)$, 因此有

$$\tilde{u}^k \in \Omega, \quad (u - \tilde{u}^k)^T \{\tilde{u}^k - [u^k - \beta_k F(u^k)]\} \geq 0, \quad \forall u \in \Omega. \tag{1.3}$$

进而得到

$$\tilde{u}^k \in \Omega, \quad (u - \tilde{u}^k)^T \beta_k F(u^k) \geq (u - \tilde{u}^k)^T (u^k - \tilde{u}^k), \quad \forall\, u \in \Omega. \tag{1.4}$$

**三个基本不等式** 设 $u^*$ 是变分不等式 VI$(\Omega, F)$ 的解. 由于 $\tilde{u}^k \in \Omega$, 根据变分不等式的定义(0.1), 我们有

$$\text{(FI1)} \qquad (\tilde{u}^k - u^*)^T \beta_k F(u^*) \geq 0. \tag{1.5}$$

由于 $u^* \in \Omega$, 将 (1.3) 中的任意 $u \in \Omega$ 置为 $u^*$, 则有

$$\text{(FI2)} \qquad (\tilde{u}^k - u^*)^T \big([u^k - \tilde{u}^k] - \beta_k F(u^k)\big) \geq 0. \tag{1.6}$$

根据单调算子的性质, 有

$$\text{(FI3)} \qquad (\tilde{u}^k - u^*)^T \big(\beta_k F(\tilde{u}^k) - \beta_k F(u^*)\big) \geq 0. \tag{1.7}$$

我们把 (1.5), (1.6) 和 (1.7) 叫做三个基本不等式. 投影收缩算法的寻查方向都是从这些基本不等式导出的.

# 2 求解线性单调变分不等式的一对孪生方法

如果 (0.1) 中的 $F(u) = Mu + q$, 这样的变分不等式称为线性变分不等式:

$$u^* \in \Omega, \quad (u - u^*)^T (Mu^* + q) \ge 0, \quad \forall u \in \Omega.$$

此时的单调性要求矩阵 $M^T + M$ 对称半正定, 并不要求 $M$ 对称.

## 2.1 根据预测点产生的距离函数上升方向

对线性变分不等式, 我们固定参数 $\beta$,

- **基于 (FI1+FI2) 的上升方向**. 将三个基本不等式中的前两个 (1.5) 和(1.6) 相加, 并利用 $F(u) = Mu + q$, 得到

$$\{(u^k - u^*) - (u^k - \tilde{u}^k)\}^T \{(u^k - \tilde{u}^k) - \beta M(u^k - u^*)\} \ge 0.$$



由单调性, $(u^k - u^*)^T M(u^k - u^*) \ge 0$, 从上式得到

$$(u^k - u^*)^T \underline{(I + \beta M^T)(u^k - \tilde{u}^k)} \ge \|u^k - \tilde{u}^k\|^2. \tag{2.8}$$

因此, $(I + \beta M^T)(u^k - \tilde{u}^k)$ 是未知距离函数 $\frac{1}{2}\|u - u^*\|^2$ 在 $u^k$ 处欧氏模的上升方向.

- **基于 FI1 的上升方向**. 因为 $F(u) = Mu + q$, 基本不等式 (1.5) 可以写成

$$\{(u^k - u^*) - (u^k - \tilde{u}^k)\}^T \beta\{(Mu^k + q) - M(u^k - u^*)\} \ge 0,$$

利用单调性, $(u^k - u^*)^T M(u^k - u^*) \ge 0$, 从上式得到

$$(u^k - u^*)^T \underline{\beta[M^T(u^k - \tilde{u}^k) + (Mu^k + q)]} \ge (u^k - \tilde{u}^k)^T \beta(Mu^k + q), \tag{2.9}$$

其中如果 $u^k \in \Omega$, 在 (1.3) 中利用 $F(u) = Mu + q$, 就有

$$(u^k - \tilde{u}^k)^T \beta_k (Mu^k + q) \ge \|u^k - \tilde{u}^k\|^2.$$

因此, 对 $u^k \in \Omega$, $\beta_k\big(M^T(u^k - \tilde{u}^k) + (Mu^k + q)\big)$ 是未知距离函数 $\frac{1}{2}\|u - u^*\|^2$ 在 $u^k$ 处欧氏模的上升方向.

**孪生方向**. 对线性变分不等式, (1.4) 可以写成

$$\tilde{u}^k \in \Omega, \quad (u-\tilde{u}^k)^T\beta(Mu^k+q) \ge (u-\tilde{u}^k)^T(u^k-\tilde{u}^k), \quad \forall\, u \in \Omega.$$

两边都加上 $(u-\tilde{u}^k)^T\beta M^T(u^k-\tilde{u}^k)$, 就有

$$\begin{array}{ll}\tilde{u}^k \in \Omega, & (u-\tilde{u}^k)^T\underline{\beta[M^T(u^k-\tilde{u}^k)+(Mu^k+q)]} \\ & \qquad \ge (u-\tilde{u}^k)^T\underline{(I+\beta M^T)(u^k-\tilde{u}^k)}, \quad \forall u \in \Omega, \end{array} \tag{2.10}$$

我们称分处 (2.10) 两端的

$$\beta[M^T(u^k-\tilde{u}^k)+(Mu^k+q)] \quad \text{和} \quad (I+\beta M^T)(u^k-\tilde{u}^k) \tag{2.11}$$

为一对孪生方向. 它们分别是由 (FI1) 和 (FI1+FI2) 产生的. 为了方便, 我们记

$$g(u^k,\tilde{u}^k) = M^T(u^k-\tilde{u}^k)+(Mu^k+q). \tag{2.12}$$

利用这个表达式, (2.10)

$$\tilde{u}^k \in \Omega, \quad (u-\tilde{u}^k)^T\underline{\beta g(u^k,\tilde{u}^k)} \ge (u-\tilde{u}^k)^T\underline{(I+\beta M^T)(u^k-\tilde{u}^k)}, \quad \forall u \in \Omega. \tag{2.13}$$



## 2.2 利用 (FI1+FI2) 提供的方向产生新的迭代点

校正是利用距离函数的下降方向(上升方向的反方向), 使得新的迭代点离解集更近一些. 利用 (FI1+FI2) 提供的方向产生新迭代点的公式为

$$u_I^{k+1}(\alpha) = u^k - \alpha(I+\beta M^T)(u^k-\tilde{u}^k). \tag{2.14}$$

为讨论步长 $\alpha$ 如何取, 我们将 (2.14) 产生的依赖于步长 $\alpha$ 的新迭代点记为 $u_I^{k+1}(\alpha)$. 考察与 $\alpha$ 相关的距离平方缩短量,

$$\vartheta_k(\alpha) := \|u^k-u^*\|^2 - \|u_I^{k+1}(\alpha)-u^*\|^2. \tag{2.15}$$

根据定义

$$\begin{aligned}\vartheta_k(\alpha) &= \|u^k-u^*\|^2 - \|u^k-u^*-\alpha(I+\beta M^T)(u^k-\tilde{u}^k)\|^2 \\ &= 2\alpha(u^k-u^*)^T(I+\beta M^T)(u^k-\tilde{u}^k) \\ &\quad -\alpha^2\|(I+\beta M^T)(u^k-\tilde{u}^k)\|^2. \end{aligned} \tag{2.16}$$

对任意给定的确定解点 $u^*$, (2.16) 表明 $\vartheta_k(\alpha)$ 是 $\alpha$ 的一个二次函数. 只是 $u^*$ 是未知的, 我们无法直接求 $\vartheta_k(\alpha)$ 的极大. 借助 (2.8), 我们有

**Theorem 2.1** 设 $u^{k+1}(\alpha)$ 由 (2.14) 生成. 对任意的 $\alpha > 0$, 由 (2.15) 定义的 $\vartheta_k(\alpha)$ 有

$$\vartheta_k(\alpha) \ge q_k^L(\alpha), \tag{2.17}$$

其中

$$q_k^L(\alpha) = 2\alpha\|u^k - \tilde{u}^k\|^2 - \alpha^2\|(I + \beta M^T)(u^k - \tilde{u}^k)\|^2. \tag{2.18}$$

**证明**. 这个结论可以从 (2.16) 利用 (2.8) 直接得到. □

定理 2.1 表明二次函数 $q_k(\alpha)$ 是 $\vartheta_k(\alpha)$ 的一个下界函数. 使 $q_k^L(\alpha)$ 达到极大的 $\alpha_k^*$ 是

$$\alpha_k^* = \text{argmax}\{q_k^L(\alpha)\} = \frac{\|u^k - \tilde{u}^k\|^2}{\|(I + \beta M^T)(u^k - \tilde{u}^k)\|^2}. \tag{2.19}$$

注意到

$$\alpha_k^* \ge \frac{1}{\|I + \beta M^T\|^2}.$$

收缩算法的本意是想在每次迭代中极大化二次函数 $\vartheta_k(\alpha)$ (见 (2.16)), 由于它含有未知的 $u^*$, 我们不得已才极大化它的下界函数 $q_k^L(\alpha)$.



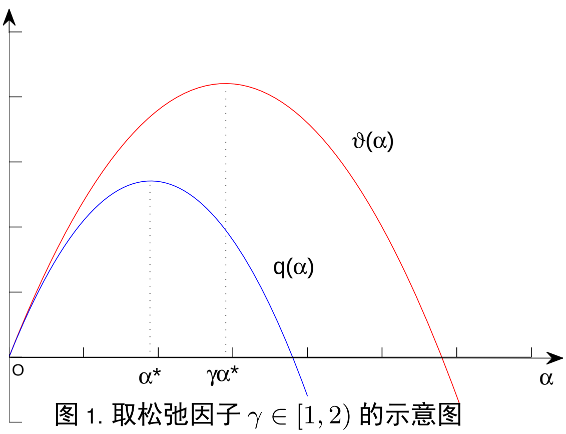



图 1. 取松弛因子 $\gamma \in [1, 2)$ 的示意图

因此, 在实际计算中, 我们一般取一个松弛因子 $\gamma \in [1, 2)$, 令

$$u^{k+1} = u^k - \gamma\alpha_k^*(I + \beta M^T)(u^k - \tilde{u}^k), \tag{2.20}$$

取 $\gamma \in [1, 2)$ 的理由可见相应的示意图. 根据 (2.15) 和 (2.17), 由 (2.20) 产生的

$u^{k+1}$ 满足

$$\|u^{k+1}-u^*\|^2 \le \|u^k-u^*\|^2 - q_k^L(\gamma\alpha_k^*).$$

由 $q_k^L(\alpha)$ 和 $\alpha_k^*$ 的定义 (分别见 (2.18) 和 (2.19)), 得到

$$\begin{aligned} q_k^L(\gamma\alpha_k^*) &= 2\gamma\alpha_k^*\|u^k-\tilde{u}^k\|^2-\gamma^2(\alpha_k^*)^2\|(I+\beta M^T)(u^k-\tilde{u}^k)\|^2 \\ &= \gamma(2-\gamma)\alpha_k^*\|u^k-\tilde{u}^k\|^2. \end{aligned}$$

因此, 由校正公式 (2.20) 产生的 $u^{k+1}$ 满足

$$\|u^{k+1}-u^*\|^2 \le \|u^k-u^*\|^2 - \frac{\gamma(2-\gamma)}{\|I+\beta M^T\|^2}\|u^k-\tilde{u}^k\|^2. \tag{2.21}$$

## 2.3 利用 (FI1) 提供的方向产生新的迭代点

§2.2 中的校正公式 (2.14) 采用 $(I+\beta M^T)(u^k-\tilde{u}^k)$ 为搜索方向, 这里我们用它的孪生方向 (见 (2.11))

$$\beta[M^T(u^k-\tilde{u}^k)+(Mu^k+q)]$$



替代它. 注意到 §2.1 的分析中说到上述方向对 $u^k\in\Omega$ 是上升的, 我们用

$$u_{II}^{k+1}(\alpha)=P_\Omega\{u^k-\alpha\beta[M^T(u^k-\tilde{u}^k)+(Mu^k+q)]\}, \tag{2.22}$$

产生依赖于步长 $\alpha$ 的新的迭代点, 也保证属于 $\Omega$. 为了区别与由 (2.14) 生成的新迭代点, 我们在 (2.22) 中用带下标的 $u_{II}^{k+1}(\alpha)$ 表示. 对任意给定的 $u^*\in\Omega^*$, 我们将

$$\zeta_k(\alpha)=\|u^k-u^*\|^2-\|u_{II}^{k+1}(\alpha)-u^*\|^2 \tag{2.23}$$

看成是本次迭代的进步量, 它是步长 $\alpha$ 的函数. 我们不能直接极大化 $\zeta_k(\alpha)$, 因为它含有我们要求的 $u^*$. 下面的定理说明, 对同样的 $\alpha$, $\zeta_k(\alpha)$ '优于' (2.17) 中的 $\vartheta_k(\alpha)$.

完成这个证明历时好几年, 证明不出就怀疑是个错误的猜想, 放在一旁.

**Theorem 2.2** 设 $u^{k+1}(\alpha)$ 由 (2.22) 生成. 对任意的 $\alpha>0$, 由 (3.18) 定义的 $\zeta_k(\alpha)$ 有

$$\zeta_k(\alpha)\ge q_k^L(\alpha)+\|u_{II}^{k+1}(\alpha)-u_I^{k+1}(\alpha)\|^2, \tag{2.24}$$

其中 $q_k^L(\alpha), u_I^{k+1}(\alpha)$, $u_{II}^{k+1}(\alpha)$ 分别由 (2.18), (2.14) 和 (2.22) 给出.

**证明**. 利用 (2.12) 的记号, (2.22) 可以写成 $u_{II}^{k+1}(\alpha)=P_\Omega[u-\alpha\beta g(u^k,\tilde{u}^k)]$.

由于 $u^* \in \Omega$, 投影算子是非扩张算子, 我们有

$$\|u_{II}^{k+1}(\alpha) - u^*\|^2 \;\le\; \|u^k - \alpha\beta g(u^k, \tilde{u}^k) - u^*\|^2.$$

然而, 根据投影的性质和余弦定理, 我们用更精细的关系式

$$\begin{aligned}\|u_{II}^{k+1}(\alpha) - u^*\|^2 \quad &\le \quad \|u^k - \alpha\beta g(u^k, \tilde{u}^k) - u^*\|^2 \\ &\quad -\|u_{II}^{k+1}(\alpha) - (u^k - \alpha\beta g(u^k, \tilde{u}^k))\|^2. \quad \text{(2.25)}\end{aligned}$$

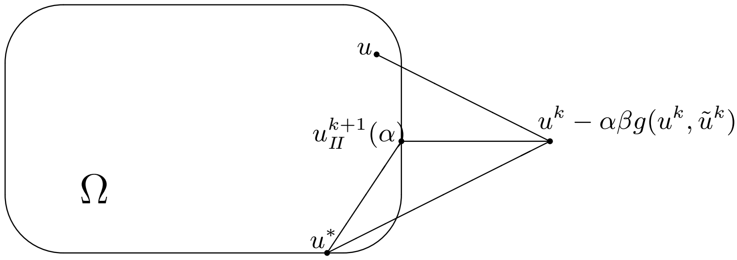



图 2. 不等式 (2.25) 的几何解释



因此, 利用 $\zeta_k(\alpha)$ 的定义(见 (2.23))和 (2.25), 我们有

$$\begin{aligned}\zeta_k(\alpha) \quad &\ge \quad \|u^k - u^*\|^2 - \|(u^k - u^*) - \alpha\beta g(u^k, \tilde{u}^k)\|^2 \\ &\qquad +\|(u_{II}^{k+1}(\alpha) - u^k) + \alpha\beta g(u^k, \tilde{u}^k)\|^2 \\ &= \quad 2\alpha\beta(u^k - u^*)^T g(u^k, \tilde{u}^k) + 2\alpha\beta(u_{II}^{k+1}(\alpha) - u^k)^T g(u^k, \tilde{u}^k) \\ &\qquad +\|u_{II}^{k+1}(\alpha) - u^k\|^2 \\ &= \quad \|u_{II}^{k+1}(\alpha) - u^k\|^2 + 2\alpha(u_{II}^{k+1}(\alpha) - u^*)^T \beta g(u^k, \tilde{u}^k). \qquad \text{(2.26)}\end{aligned}$$

将 (2.26) 中右端的最后一项 $(u_{II}^{k+1}(\alpha) - u^*)^T \beta g(u^k, \tilde{u}^k)$ 分解成

$$\begin{aligned}&(u_{II}^{k+1}(\alpha) - u^*)^T \beta g(u^k, \tilde{u}^k) \\ &\quad = \quad (u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T \beta g(u^k, \tilde{u}^k) + (\tilde{u}^k - u^*)^T \beta g(u^k, \tilde{u}^k). \quad \text{(2.27)}\end{aligned}$$

对于 (2.27) 右端的第一部分, 利用记号 $g(u^k, \tilde{u}^k)$, $u_{II}^{k+1}(\alpha) \in \Omega$ 和 (2.10), 有

$$(u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T \beta g(u^k, \tilde{u}^k) \ge (u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T (I + \beta M^T)(u^k - \tilde{u}^k).$$

也就是说,

$$
\begin{aligned}
(u_{II}^{k+1}(\alpha)-\tilde{u}^k)^T\beta g(u^k,\tilde{u}^k) \quad \ge \quad & (u_{II}^{k+1}(\alpha)-u^k)^T(I+\beta M^T)(u^k-\tilde{u}^k) \\
& +(u^k-\tilde{u}^k)^T(I+\beta M^T)(u^k-\tilde{u}^k). \quad (2.28)
\end{aligned}
$$

对于 (2.27) 右端的第二部分, $(\tilde{u}^k-u^*)^T\beta g(u^k,\tilde{u}^k)$, 再分拆成

$$
(\tilde{u}^k-u^*)^T\beta g(u^k,\tilde{u}^k)=(\tilde{u}^k-u^k)^T\beta g(u^k,\tilde{u}^k)+(u^k-u^*)^T\beta g(u^k,\tilde{u}^k)
$$

利用 (2.9), 即

$$
(u^k-u^*)^T\beta g(u^k,\tilde{u}^k)\ge(u^k-\tilde{u}^k)^T(Mu^k+q)
$$

和记号 $g(u^k,\tilde{u}^k)$, 我们得到

$$
\begin{aligned}
&(\tilde{u}^k-u^*)^T\beta g(u^k,\tilde{u}^k) \\
&\quad \ge \quad (\tilde{u}^k-u^k)^T\beta\{M^T(u^k-\tilde{u}^k)+(Mu^k+q)\}+(u^k-\tilde{u}^k)^T(Mu^k+q) \\
&\quad \ge \quad -\beta(u^k-\tilde{u}^k)^TM^T(u^k-\tilde{u}^k). \qquad (2.29)
\end{aligned}
$$

将 (2.28) 和 (2.29) 相加, 就有

$$
(u_{II}^{k+1}(\alpha)-u^*)^T\beta g(u^k,\tilde{u}^k)\ge(u_{II}^{k+1}(\alpha)-\tilde{u}^k)^T(I+\beta M^T)(u^k-\tilde{u}^k)+\|u^k-\tilde{u}^k\|^2.
$$



将上面的不等式代入 (2.26), 利用 $q_k^L(\alpha)$ 的记号, 得到

$$
\begin{aligned}
\zeta_k(\alpha) \quad \ge \quad & \|u_{II}^{k+1}(\alpha)-u^k\|^2+2\alpha(u_{II}^{k+1}(\alpha)-u^k)^T(I+\beta M^T)(u^k-\tilde{u}^k) \\
& +2\alpha\|u^k-\tilde{u}^k\|^2 \\
= \quad & \|(u_{II}^{k+1}(\alpha)-u^k)+\alpha(I+\beta M^T)(u^k-\tilde{u}^k)\|^2 \\
& -\alpha^2\|(I+\beta M^T)(u^k-\tilde{u}^k)\|^2+2\alpha\|u^k-\tilde{u}^k\|^2 \\
= \quad & \|u_{II}^{k+1}(\alpha)-[u^k-\alpha(I+\beta M^T)(u^k-\tilde{u}^k)]\|^2+q_k^L(\alpha).
\end{aligned}
$$

再利用 (2.14), 就完成了定理结论 (2.24) 的证明. □

我清楚记得, 这个证明 [8] 是 1991 年的农历中秋节晚上在德国完成的.

定理 2.2 说明, $q_k^L(\alpha)$ 也是 $\zeta_k(\alpha)$ 的下界. 在实际计算中, 无论采用校正公式

$$
\text{(PC-I)} \quad u_I^{k+1}=u^k-\gamma\alpha_k^*(I+\beta M^T)(u^k-\tilde{u}^k) \qquad (2.30)
$$

或者

$$
\text{(PC-II)} \quad u_{II}^{k+1}=P_\Omega\big[u^k-\gamma\alpha_k^*\beta[M^T(u^k-\tilde{u}^k)+(Mu^k+q)]\big] \qquad (2.31)
$$

产生新的迭代点 $u^{k+1}$, 其中的 $\alpha_k^*$ 都由 (2.19) 给出. 步长下有界, 这点非常重要. 一个有效的迭代算法, 步长必须是下有界的.

采用校正公式 (2.30), 它的好处是生成 $u^{k+1}$ 不用再做投影. 实际问题中, 到 $\Omega$ 上的投影代价往往不高（例如 $\Omega$ 常常是一个正卦限或者框形), 因此常采用校正公式 (2.31). 这方面的理由我们在论文 [14] 中有更详细的说明.

根据 (2.10) 提供的孪生方向, 在采用相同步长的两个方法 (2.30) 和 (2.31) 中, 用记号 (2.14), (2.17) 和 (2.24) 就可以分别改写成

$$\vartheta_k(\alpha) = \|u^k - u^*\|^2 - \|u_I^{k+1}(\alpha) - u^*\|^2 \ \geq \ q_k^L(\alpha),$$

和

$$\zeta_k(\alpha) = \|u^k - u^*\|^2 - \|u_{II}^{k+1}(\alpha) - u^*\|^2 \ \geq \ q_k^L(\alpha) + \|u_I^{k+1}(\alpha) - u_{II}^{k+1}(\alpha)\|^2.$$

其中 $q_k^L(\alpha)$ 由 (2.18) 给出. 对同样的 $\alpha$, $\zeta_k(\alpha)$ 的下界大于 $\vartheta_k(\alpha)$ 的下界.

求解线性变分不等式的 PC Method-I 和 PC Method-II 分别发表在 [6] 和 [7], 都被应用到机器人的运动规划和实时控制中 [5, 16]. 我们在 1994 年发表的 PC Method-I , 文章 [6] 的题目是 “A new method for a class of linear variational inequalities”, 引用人称它为 **94LVI** 算法. 同样是在1994 年发表的论文 [7] 中报导 PC Method-II 的公式编号是 (4) 到 (7), 引用人就给这个算法冠名为 **E47**.

他们的计算实践 [5, 16] 也证明, 这一对孪生算法中, 算法-II 比算法-I 效率要高.



# 3 非线性单调变分不等式的一对孪生方法

对一般的非线性单调变分不等式, 在投影 (1.2) 中, 我们假设参数 $\beta_k$ 选的能使

$$\beta_k\|F(u^k) - F(\tilde{u}^k)\| \leq \nu\|u^k - \tilde{u}^k\|, \quad \nu \in (0,1). \tag{3.1}$$

后面, 为了行文简单, 我们将 $\beta_k$ 记成 $\beta$, 并假设这个 $\beta$ 满足上面的条件.

## 3.1 根据预测点产生的距离函数上升方向

- **基于 (FI1+FI2+FI3) 的上升方向**. 将三个基本不等式 (1.5), (1.6) 和(1.7) 相加, 将三个基本不等式 (1.5), (1.6) 和(1.7) 相加, 并利用 $d(u^k, \tilde{u}^k)$ 的表达式, 得到

$$\{(u^k - u^*) - (u^k - \tilde{u}^k)\}^T d(u^k, \tilde{u}^k) \geq 0,$$

其中

$$d(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k) - \beta[F(u^k) - F(\tilde{u}^k)]. \tag{3.2}$$

随后得到

$$(u^k - u^*)^T d(u^k, \tilde{u}^k) \geq (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k). \tag{3.3}$$

由 $d(u^k,\tilde{u}^k)$ 的表达式 (3.2) 和假设 (3.1), 根据 Cauchy-Schwarz 不等式, 有

$$(u^k-\tilde{u}^k)^T d(u^k,\tilde{u}^k) \ge (1-\nu)\|u^k-\tilde{u}^k\|^2. \tag{3.4}$$

不等式 (3.3) 和 (3.4) 告诉我们, 在条件 (3.1) 满足的情况下, 由 (3.2) 定义的 $d(u^k,\tilde{u}^k)$ 是欧氏模下的一个上升方向.

- **基于 (FI1+ FI3) 的上升方向**. 将基本不等式 (1.5) 可以写成

$$\{(u^k-u^*)-(u^k-\tilde{u}^k)\}^T \beta F(\tilde{u}^k) \ge 0,$$

从上式得到

$$(u^k-u^*)^T\beta F(\tilde{u}^k) \ge (u^k-\tilde{u}^k)^T\beta F(\tilde{u}^k). \tag{3.5}$$

如果 $u^k\in\Omega$, 根据 (1.4) 有

$$(u^k-\tilde{u}^k)^T\beta F(u^k) \ge \|u^k-\tilde{u}^k\|^2.$$

利用上式, 假设 (3.1) 和 Cauchy-Schwarz 不等式, 有

$$\begin{aligned}(u^k-\tilde{u}^k)^T\beta F(\tilde{u}^k) &= (u^k-\tilde{u}^k)^T\beta F(u^k)-(u^k-\tilde{u}^k)^T\beta(F(u^k)-F(\tilde{u}^k))\\ &\ge (1-\nu)\|u^k-\tilde{u}^k\|^2. \end{aligned}\tag{3.6}$$



不等式 (3.5) 和 (3.6) 告诉我们, 对 $u^k\in\Omega$, 在条件 (3.1) 满足的情况下, $\beta F(\tilde{u}^k)$ 是未知距离函数 $\frac{1}{2}\|u-u^*\|^2$ 在 $u^k$ 处欧氏模的上升方向.

**孪生方向**. 考察上面提到的两个方向的关系. 在 (1.4) 两边都加上

$$(u-\tilde{u}^k)^T\{-\beta[F(u^k)-F(\tilde{u}^k)]\},$$

就有

$$\tilde{u}^k\in\Omega,\quad (u-\tilde{u}^k)^T\underline{\beta F(\tilde{u}^k)} \ge (u-\tilde{u}^k)^T\underline{d(u^k,\tilde{u}^k)},\quad \forall u\in\Omega, \tag{3.7}$$

我们称分处 (3.7) 两端的

$$\beta F(\tilde{u}^k) \qquad 和 \qquad d(u^k,\tilde{u}^k)=(u^k-\tilde{u}^k)-\beta[F(u^k)-F(\tilde{u}^k)]$$

为一对孪生方向, 它们的分别是由 (FI1+FI3) 和 (FI1 +FI2 +FI3) 产生的.

## 3.2 利用 (FI1+FI2+FI3) 提供的方向产生新的迭代点

校正是利用距离函数的下降方向(上升方向的反方向), 使得新的迭代点离解

集更近一些. 因此, 我们用

$$u_I^{k+1} = u^k - \alpha d(u^k, \tilde{u}^k) \tag{3.8}$$

产生新的迭代点, 其中 $d(u^k, \tilde{u}^k)$ 是由 (3.2) 给出的.

我们将 (3.8) 中的 $u^{k+1}$ 记为 $u^{k+1}(\alpha)$, 表示新的迭代点依赖于步长 $\alpha$. 考察与 $\alpha$ 相关的距离平方缩短量,

$$\vartheta_k(\alpha) := \|u^k - u^*\|^2 - \|u_I^{k+1}(\alpha) - u^*\|^2. \tag{3.9}$$

根据定义

$$\begin{aligned} \vartheta_k(\alpha) &= \|u^k - u^*\|^2 - \|u^k - u^* - \alpha d(u^k, \tilde{u}^k)\|^2 \\ &= 2\alpha(u^k - u^*)^T d(u^k, \tilde{u}^k) - \alpha^2 \|d(u^k, \tilde{u}^k)\|^2. \end{aligned} \tag{3.10}$$

对任意给定的确定解点 $u^*$, (3.10) 表明 $\vartheta_k(\alpha)$ 是 $\alpha$ 的一个二次函数. 只是 $u^*$ 是未知的, 我们无法直接求 $\vartheta_k(\alpha)$ 的极大.

**Theorem 3.1** 设 $u^{k+1}(\alpha)$ 由 (3.8) 生成. 对任意的 $\alpha > 0$, 由 (3.9) 定义的 $\vartheta_k(\alpha)$ 有

$$\vartheta_k(\alpha) \geq q_k^N(\alpha), \tag{3.11}$$



其中 $d(u^k, \tilde{u}^k)$ 由 (3.2) 给出,

$$q_k^N(\alpha) = 2\alpha(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k) - \alpha^2 \|d(u^k, \tilde{u}^k)\|^2. \tag{3.12}$$

**证明**. 这个结论可以从 (3.10) 利用 (3.3) 直接得到. □

定理 3.1 表明二次函数 $q_k^N(\alpha)$ 是 $\vartheta_k(\alpha)$ 的一个下界函数. 使 $q_k^N(\alpha)$ 达到极大的 $\alpha_k^*$ 是

$$\alpha_k^* = \text{argmax}\{q_k^N(\alpha)\} = \frac{(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k)}{\|d(u^k, \tilde{u}^k)\|^2}. \tag{3.13}$$

收缩算法的本意是想在每次迭代中极大化二次函数 $\vartheta_k(\alpha)$ (见 (3.10)), 由于它含有未知的 $u^*$, 我们不得已才极大化它的下界函数 $q_k^N(\alpha)$.

因此, 在实际计算中, 我们一般取一个松弛因子 $\gamma \in [1, 2)$, 令

$$u^{k+1} = u^k - \gamma \alpha_k^* d(u^k, \tilde{u}^k), \tag{3.14}$$

取 $\gamma \in [1, 2)$ 的理由可见相应的示意图 $1$. 根据 (3.9) 和 (3.11), 由 (3.14) 产生的 $u^{k+1}$ 满足

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - q_k^N(\gamma \alpha_k^*).$$

由 $q_k^N(\alpha)$ 和 $\alpha_k^*$ 的定义 (分别见 (3.12) 和 (3.13)), 得到

$$\begin{aligned} q_k^N(\gamma\alpha_k^*) &= 2\gamma\alpha_k^*(u^k-\tilde{u}^k)^T d(u^k,\tilde{u}^k) - \gamma^2(\alpha_k^*)^2\|d(u^k,\tilde{u}^k)\|^2 \\ &= \gamma(2-\gamma)\alpha_k^*(u^k-\tilde{u}^k)^T d(u^k,\tilde{u}^k). \end{aligned}$$

事实上, 利用 (3.2) 和 (3.1), 可以得到

$$\begin{aligned} &2(u^k-\tilde{u}^k)^T d(u^k,\tilde{u}^k) - \|d(u^k,\tilde{u}^k)\|^2 = d(u^k,\tilde{u}^k)^T\{2(u^k-\tilde{u}^k) - d(u^k,\tilde{u}^k)\} \\ &\quad= \big\{(u^k-\tilde{u}^k) - \beta[F(u^k)-F(\tilde{u}^k)]\big\}^T\big\{(u^k-\tilde{u}^k) + \beta[F(u^k)-F(\tilde{u}^k)]\big\} \\ &\quad= \|u^k-\tilde{u}^k\|^2 - \beta_k^2\|[F(u^k)-F(\tilde{u}^k)]\|^2 \;\geq\; (1-\nu^2)\|u^k-\tilde{u}^k\|^2. \end{aligned} \tag{3.15}$$

因此, $\alpha_k^* > \frac{1}{2}$. 由校正公式 (3.14) 产生的 $u^{k+1}$ 满足

$$\|u^{k+1}-u^*\|^2 \leq \|u^k-u^*\|^2 - \frac{\gamma(2-\gamma)}{2}(u^k-\tilde{u}^k)^T d(u^k,\tilde{u}^k). \tag{3.16}$$

注意到, 上式中的 $(u^k-\tilde{u}^k)^T d(u^k,\tilde{u}^k)$ 是 (3.3) 的右端, 利用 (3.4) 有

$$\|u^{k+1}-u^*\|^2 \leq \|u^k-u^*\|^2 - \frac{\gamma(2-\gamma)}{2}(1-\nu)\|u^k-\tilde{u}^k\|^2.$$



## 3.3 利用 (FI1+FI3) 提供的方向产生新的迭代点

§3.2 的校正公式 (3.8) 中采用 $d(u^k,\tilde{u}^k)$ 为方向, 这里我们用 $\beta F(\tilde{u}^k)$ 替代它, 并附加一次投影. 同样用

$$u_{II}^{k+1}(\alpha) = P_\Omega[u^k - \alpha\beta F(\tilde{u}^k)], \tag{3.17}$$

为依赖于步长 $\alpha$ 的新的迭代点. 为了区别与由 (3.8) 生成的 $u^{k+1}$, 我们在 (3.17) 中用带下标的 $u_{II}^{k+1}(\alpha)$ 表示. 对任意给定的 $u^* \in \Omega^*$, 我们将

$$\zeta_k(\alpha) = \|u^k-u^*\|^2 - \|u_{II}^{k+1}(\alpha)-u^*\|^2 \tag{3.18}$$

看成是本次迭代的进步量, 它是步长 $\alpha$ 的函数. 我们不能直接极大化 $\zeta_k(\alpha)$, 因为它含有我们要求的 $u^*$. 下面的定理说明, 对同样的 $\alpha$, $\zeta_k(\alpha)$ '优于' (3.11) 中的 $\vartheta_k(\alpha)$.

**Theorem 3.2** 设 $u^{k+1}(\alpha)$ 由 (3.17) 生成. 对任意的 $\alpha > 0$, 由 (3.18) 定义的 $\zeta_k(\alpha)$ 有

$$\zeta_k(\alpha) \geq q_k^N(\alpha) + \|u_{II}^{k+1}(\alpha) - u_I^{k+1}(\alpha)\|^2, \tag{3.19}$$

其中 $q_k^N(\alpha), u_I^{k+1}(\alpha)$, $u_{II}^{k+1}(\alpha)$ 分别由 (3.12), (3.8) 和 (3.17) 给出.

**证明**. 首先, 因为 $u^{k+1}(\alpha) = P_\Omega[u - \alpha\beta F(\tilde{u}^k)]$ 和 $u^* \in \Omega$, 根据投影的性质和余弦定理, 有

$$\|u^{k+1}(\alpha) - u^*\|^2 \leq \|u^k - \alpha\beta F(\tilde{u}^k) - u^*\|^2 - \|u^k - \alpha\beta F(\tilde{u}^k) - u_{II}^{k+1}(\alpha)\|^2. \tag{3.20}$$

因此, 利用 $\zeta_k(\alpha)$ 的定义(见 (3.18)), 我们有

$$\begin{aligned} \zeta_k(\alpha) &\geq \|u^k - u^*\|^2 - \|(u^k - u^*) - \alpha\beta F(\tilde{u}^k)\|^2 + \|(u^k - u_{II}^{k+1}(\alpha)) - \alpha\beta F(\tilde{u}^k)\|^2 \\ &= 2\alpha(u^k - u^*)^T\beta F(\tilde{u}^k) + 2\alpha(u_{II}^{k+1}(\alpha) - u^k)^T\beta F(\tilde{u}^k) + \|u^k - u_{II}^{k+1}(\alpha)\|^2 \\ &= \|u^k - u_{II}^{k+1}(\alpha)\|^2 + 2\alpha(u_{II}^{k+1}(\alpha) - u^*)^T\beta F(\tilde{u}^k). \end{aligned} \tag{3.21}$$

将 (3.21) 中右端的最后一项 $(u_{II}^{k+1}(\alpha) - u^*)^T\beta F(\tilde{u}^k)$ 分解成

$$(u_{II}^{k+1}(\alpha) - u^*)^T\beta F(\tilde{u}^k) = (u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T\beta F(\tilde{u}^k) + (\tilde{u}^k - u^*)^T\beta F(\tilde{u}^k).$$

利用 $F$ 的单调性和 $u^*$ 的最优性,

$$(\tilde{u}^k - u^*)^T\beta F(\tilde{u}^k) \geq (\tilde{u}^k - u^*)^T\beta F(u^*) \geq 0.$$

代入 (3.21) 的右端, 进一步得到

$$\zeta_k(\alpha) = \|u_{II}^{k+1}(\alpha) - u^k\|^2 + 2\alpha(u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T\beta_k F(\tilde{u}^k). \tag{3.22}$$



因为 $u_{II}^{k+1}(\alpha) \in \Omega$, 用它替代 (3.7) 中的任意 $u \in \Omega$, 得到

$$(u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T\beta_k F(\tilde{u}^k) \geq (u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T d(u^k, \tilde{u}^k).$$

将它代入 (3.22) 的右端, 就有

$$\zeta_k(\alpha) \geq \|u_{II}^{k+1}(\alpha) - u^k\|^2 + 2\alpha(u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T d(u^k, \tilde{u}^k). \tag{3.23}$$

对上式右端, 利用 $q_k^N(\alpha)$ 的形式 (见(3.12)), 就化成

$$\begin{aligned} &\|u_{II}^{k+1}(\alpha) - u^k\|^2 + 2\alpha(u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T d(u^k, \tilde{u}^k) \\ &= \|u_{II}^{k+1}(\alpha) - u^k\|^2 + 2\alpha(u_{II}^{k+1}(\alpha) - u^k)^T d(u^k, \tilde{u}^k) + 2\alpha(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k) \\ &= \|(u_{II}^{k+1}(\alpha) - u^k) + \alpha d(u^k, \tilde{u}^k)\|^2 - \alpha^2\|d(u^k, \tilde{u}^k)\|^2 + 2\alpha(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k) \\ &= \|u_{II}^{k+1}(\alpha) - (u^k - \alpha d(u^k, \tilde{u}^k))\|^2 + q_k^N(\alpha). \end{aligned}$$

这样就完成了定理结论 (3.19) 的证明. □

定理 3.2 说明, $q_k^N(\alpha)$ 也是 $\zeta_k(\alpha)$ 的下界. 在实际计算中, 采用校正公式

$$\text{(PC Method-I)} \quad u_I^{k+1} = u^k - \gamma\alpha_k^* d(u^k, \tilde{u}^k) \tag{3.24}$$

或者

$$\text{(PC Method-II)} \quad u_{II}^{k+1} = P_\Omega[u^k - \gamma\alpha_k^*\beta_k F(\tilde{u}^k)] \tag{3.25}$$

产生新的迭代点 $u^{k+1}$, 其中的 $\alpha_k^*$ 都由 (3.13) 给出. 采用校正公式 (3.24), 它的好处是生成 $u^{k+1}$ 不用再做投影. 实际问题中, 到 $\Omega$ 上的投影代价往往不高 (例如 $\Omega$ 常常是一个正卦限或者框形), 因此常采用校正公式 (3.25).

利用§ 3.2 最后证明的结果, 无论采取孪生方法中的哪一个, 迭代产生的序列 $\{u^k\}$ 都满足

$$\|u^{k+1} - u^*\|^2 \le \|u^k - u^*\|^2 - \frac{\gamma(2-\gamma)}{2}(1-\nu)\|u^k - \tilde{u}^k\|^2.$$

据此就可以证明下面的收敛性定理.

**Theorem 3.3** 设 *VI*$(\Omega, F)$ 的解集 $\Omega^*$ 非空, 则由本节提到的投影收缩算法产生的序列 $\{u^k\}$ 收敛到 *VI*$(\Omega, F)$ 的某个解点 $u^* \in \Omega^*$.

中国科学院武汉岩土力学研究所的科研人员, 将这里介绍的投影-收缩算法成功用于许多岩土工程问题的求解 [17, 18], 他们的成果也向我们做了通报.



# 4 应用和数值试验

我们分别用线性变分不等式和非线性变分不等式的例子说明统一框架中二种方法— PC Method-I 和PC Method-II 的不同计算效果.

## 4.1 Applied the different PC Methods for LVI

线性变分不等式我们用第一讲 §5 中提到的“最短距离和问题”作为例子. 最短距离和问题等价于一个 min-max 问题, 其对应的线性变分不等式中的矩阵 $M$ 是斜对称的. 对此类问题的详细的描述见第一讲 §5.

**试验例子**: 取自 SIAM J. on Optimization.

- G. L. XUE AND Y. Y. YE, *An efficient algorithm for minimizing a sum of Euclidean norms with applications*, SIAM Optim. 7 (1997), 1017-1039.

Fig. 1 给出这个网络的联结结构, 其中 $b_{[i]}, i = 1, \ldots, 10$ 是正则点 (regular points). 这些正则点的坐标是给定的. 点 $x_{[j]}, j = 1, \ldots, 8$ 与点$b_{[i]}$ 的联结也是给定的. Fig. 2 给出了欧氏模下距离和最短时 $x_{[j]}, j = 1, \ldots, 8$ 的位置.

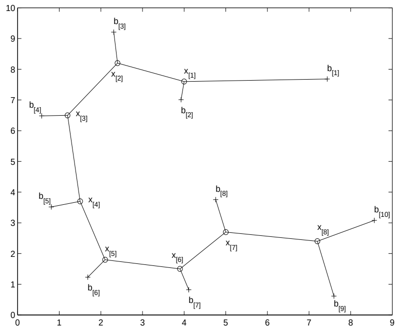



Fig. 1. Fig 1. Ordering of the topology

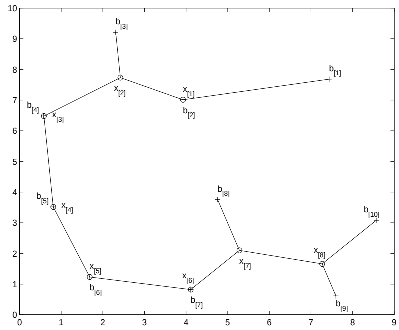



Fig. 2. Optimal solution in Euclidean-norm

**The coordinates of the 10 regular points**

| | x-coordinate | y-coordinate | | x-coordinate | y-coordinate |
|---|---|---|---|---|---|
| $b_{[1]}$ | 7.436490 | 7.683284 | $b_{[6]}$ | 1.685912 | 1.231672 |
| $b_{[2]}$ | 3.926097 | 7.008798 | $b_{[7]}$ | 4.110855 | 0.821114 |
| $b_{[3]}$ | 2.309469 | 9.208211 | $b_{[8]}$ | 4.757506 | 3.753666 |
| $b_{[4]}$ | 0.577367 | 6.480938 | $b_{[9]}$ | 7.598152 | 0.615836 |
| $b_{[5]}$ | 0.808314 | 3.519062 | $b_{[10]}$ | 8.568129 | 3.079179 |



The update forms of using the contraction method I (3.24) and II (3.25) are

$$\text{(PC Method-I)} \qquad u^{k+1} = u^k - \gamma\alpha_k^*(I + M^T)(u^k - \tilde{u}^k),$$

and

$$\text{(PC Method-II)} \qquad u^{k+1} = P_\Omega\{u^k - \gamma\alpha_k^*[M^T(u^k - \tilde{u}^k) + (Mu^k + q)]\},$$

respectively. The numerical results are listed in the following table.

Table 1. Shortest network under $l_2$ norm.

| PC Method-I | | | PC Method-II | | |
|---|---|---|---|---|---|
| Iteration | $\|e(u)\|_\infty$ | Total Distance | Iteration | $\|e(u)\|_\infty$ | Total Distance |
| 40 | 7.1e-002 | 25.3776304969 | 40 | 5.0e-004 | 25.3563526162 |
| 80 | 1.8e-004 | 25.3561050662 | 80 | 4.0e-008 | 25.3560677986 |
| 120 | 6.4e-007 | 25.3560678958 | 106 | 9.2e-011 | 25.3560677793 |
| 160 | 2.4e-009 | 25.3560677797 | | | |
| 183 | 9.5e-011 | 25.3560677793 | | | |
| CPU-time | | 0.234 Sec. | CPU-time | | 0.125 Sec. |

我们用了松弛因子 $\gamma = 1.8$. 如果取 $\gamma = 1$, 两种方法都会增加$80\%$ 的计算时间.

## PC Method-II for the problem in the Euclidean-norm

```
clear; % Steiner Minimum Tree * Read the coordinate of the regular points%(1)
 P1=[7.436490, 3.926097, 2.309469, 0.577367, 0.808314;                  %(2)
     7.683284, 7.008798, 9.208211, 6.480938, 3.519062];                 %(3)
 P2=[1.685912, 4.110855, 4.757506, 7.598152, 8.568129;                  %(4)
     1.231672, 0.821114, 3.753666, 0.615836, 3.079179];                 %(5)
 b=[P1,P2,zeros(2,7)]; x=zeros(2,8); z=zeros(2,17);  eps=1;   k=0; tic;  %(6)
while (eps > 10^(-10) & k<= 200)  k=k+1;    %% Beginning of an iteration  %(7)
 Ax=[x(:,1), x, x(:,8), x(:,1:7)-x(:,2:8)];   Axb=Ax-b;  %% Compute Ax-b %(8)
 ATz=z(:,2:9) + [z(:,1), -z(:,11:17)] + [z(:,11:17), z(:,10)];   % A^Tz  %(9)
 L2=0; for j=1:17  L2=L2 + norm(Axb(:,j),2); end;           % Length-2 %(10)
 if mod(k,20)==0 fprintf('k=%3d stopc=%9.1e L2=%13.10f\n',k,eps,L2);end;%(11)
 Pz=z+Axb; Dp=diag(1./max(1,sqrt(diag(Pz'*Pz)))); Pz=Pz*Dp; %P(z+(Ax-b))%(12)
 Ex = ATz;   Ez = z-Pz;  t=trace(Ex'*Ex)+ trace(Ez'*Ez);  eps=sqrt(t);  %(13)
 AEx= [Ex(:,1), Ex, Ex(:,8), Ex(:,1:7)-Ex(:,2:8)];      % Compute AEx   %(14)
 ATEz=Ez(:,2:9) + [Ez(:,1),-Ez(:,11:17)] + [Ez(:,11:17),Ez(:,10)]; %ATEz%(15)
  ta = trace(AEx'*AEx)+trace(ATEz'*ATEz);  alpha=t*1.8/(t+ta); %% Step L%(16)
  x =x-(ATz - ATEz)*alpha;                                 %% New x and z %%(17)
  z =z-(AEx - Axb)*alpha; Dz=diag(1./max(1,sqrt(diag(z'*z)))); z=z*Dz; %%(18)
 end;                                              %% End of an iteration %%(19)
 toc; fprintf(' k=%3d    eps=%9.1e   Length-2=%13.10f \n', k,eps,L2);  %%(20)
```

把上面第(18) 行改成 `z =z-(AEx + Ez)*alpha;`就是处理同一问题的(CM-D1)程序.



Fig. 3 and 4 depict the convergence tendencies of Contraction Method–2 for the minimum sum of the distance in the Euclidean-norm with different starting points.

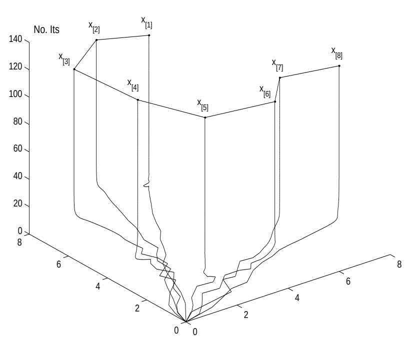



Fig. 3. Convergence tendency, $x^0 = 0$

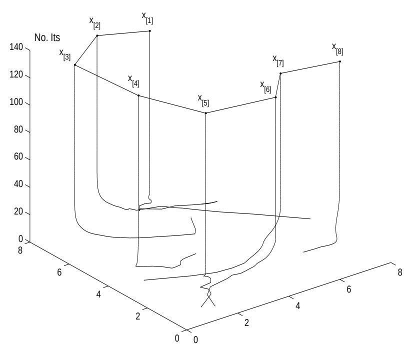



Fig. 4. Convergence tendency, $x^0$ random

对 $l_1$-模和 $l_\infty$-模, 我们也用两种不同方法做了计算比较, 计算结果如下：

Table 2. Shortest network under $l_1$ norm.

| PC Method-I | | | PC Method-II | | |
|---|---|---|---|---|---|
| Iteration | $\|e(u)\|_\infty$ | Total Distance | Iteration | $\|e(u)\|_\infty$ | Total Distance |
| 40 | 3.7e-002 | 28.6777786413 | 40 | 1.1e-004 | 28.6660178525 |
| 80 | 2.5e-005 | 28.6658649129 | 81 | 1.0e-010 | 28.6658580000 |
| 120 | 1.8e-008 | 28.6658580046 | | | |
| 149 | 9.4e-011 | 28.6658580000 | | | |
| CPU-time | | 0.031 Sec. | CPU-time | | 0.016 Sec. |

Table 3. Shortest network under $l_\infty$ norm.

| PC Method-I | | | PC Method-II | | |
|---|---|---|---|---|---|
| Iteration | $\|e(u)\|_\infty$ | Total Distance | Iteration | $\|e(u)\|_\infty$ | Total Distance |
| 40 | 9.0e-002 | 21.1322990353 | 40 | 2.1e-003 | 21.1145131146 |
| 80 | 4.4e-005 | 21.1129244226 | 80 | 4.1e-010 | 21.1129135002 |
| 120 | 2.4e-008 | 21.1129135060 | 84 | 7.4e-011 | 21.1129135000 |
| 150 | 9.2e-011 | 21.1129135000 | | | |
| CPU-time | | 0.187 Sec. | CPU-time | | 0.094 Sec. |



Fig. 5 and Fig. 6 depict the optimal solutions of the minimum sum of the distance in the $l_1$-norm and $l_\infty$-norm, respectively.

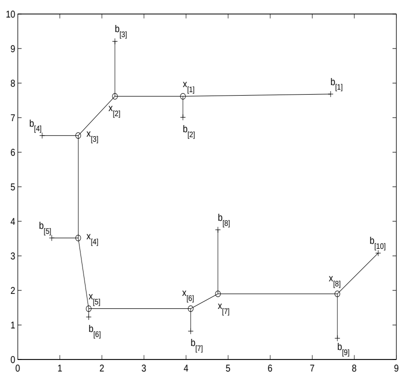


Fig. 5. Optimal solution, $l_1$-norm

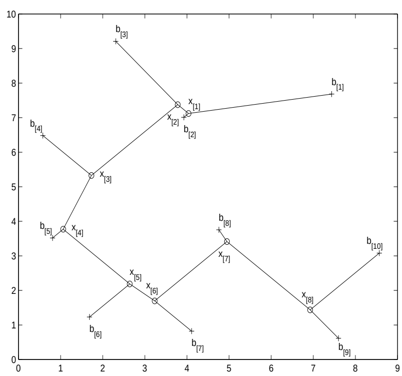


Fig. 6. Optimal solution, $l_\infty$-norm

Fig. 7 and 8 depict the convergence tendencies of Contraction Method–2 with random starting points for the minimum sum of the distance in the in the $l_1$-norm and $l_\infty$-norm, respectively.

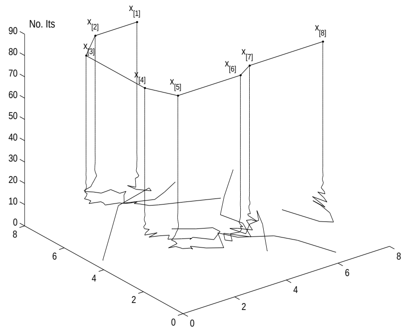



Fig. 7. Convergence tendency, $l_1$-norm

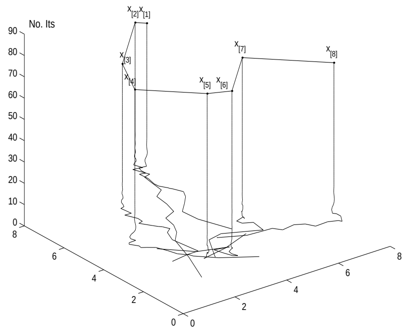



Fig. 8. Convergence tendency, $l_\infty$-norm



## 4.2 非线性互补问题

作为非线性变分不等式的例子我们用第二讲 §7 中的非线性互补问题

$$u \geq 0, \qquad F(u) \geq 0, \qquad u^T F(u) = 0,$$

来考察 PC Method-I 和 PC Method-II 的不同计算效率. 在算例中我们取

$$F(u) = D(u) + Mu + q,$$

线性部分 $Mu + q$ 采用下面的语句生成:

```
A=(rand(n,n)-0.5)*10;   B=(rand(n,n)-0.5)*10;    B=B-B';   M=A'*A+B;
q=(rand(n,1)-0.5)*1000;          或者          q=(rand(n,1)-1.0)*500;
```

非线性部分 $D(u)$ 的每个分量 $D_j(u) = d_j * \arctan(u_j)$, 其中$d_j$ 是 $(0,1)$之间的随机数. 用这一讲 §2.3 中的方法生成方向 $d_1(u,\tilde{u})$ 和 $d_2(u,\tilde{u})$, 以及误差度量函数 $\varphi(u,\tilde{u})$.

注意到, 这样做以后, PC Method-I 就是第二讲中的投影收缩算法.
置PC Method-II 中的 $\gamma\alpha_k^* \equiv 1$, 就是前一讲的 Refined 外梯度算法.

在后面的试验中, 初始迭代向量 $u^0$ 的每个分量都取 $(0,10)$ 中的随机数.

## PC Method-II 的程序

**PC Method-II:**

Step 0. Set $\beta_0 = 1, \nu \in (0,1), u^0 \in \Omega$ and $k = 0$.

Step 1. $\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)]$,

$r_k := \beta_k \|F(u^k) - F(\tilde{u}^k)\| / \|u^k - \tilde{u}^k\|$,

**while** $r_k > \nu$, $\beta_k := \frac{2}{3}\beta_k * \min\{1, \frac{1}{r_k}\}$,

$\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)]$,

$r_k := \beta_k \|F(u^k) - F(\tilde{u}^k)\| / \|u^k - \tilde{u}^k\|$,

**end(while)**

$$d(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k) - \beta_k[F(u^k) - F(\tilde{u}^k)],$$

$$\alpha_k = \frac{(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k)}{\|d(u^k, \tilde{u}^k)\|^2},$$

$$u^{k+1} = P_\Omega[u^k - \gamma\alpha_k\beta_k F(\tilde{u}^k)],$$

**If** $r_k \le \mu$ then $\beta_k := \beta_k * 1.5$, **end(if)**

Step 2. $\beta_{k+1} = \beta_k$ and $k = k+1$, go to Step 1.

从 PC Method-I 到收缩算法 PC Method-II 只是将

$u^{k+1} = u^k - \gamma\alpha_k d(u^k, \tilde{u}^k)$ 改成了 $u^{k+1} = P_\Omega[u^k - \gamma\alpha_k\beta_k F(\tilde{u}^k)]$.



## Matlab Code of Contraction Method–D2 for NCP

```
function PC_G(n,M,q,d,xstart,tol,pfq)                                    %(1)
fprintf('PC Method use Direction D1 with gamma a*  n=%4d \n',n);         %(2)
x=xstart;  Fx= d.*atan(x) + M*x + q;  stopc=norm(x-max(x-Fx,0),inf);    %(3)
beta=1;        k=0;        l=0;        tic;                              %(4)
while (stopc>tol && k<=2000)                                             %(5)
  if mod(k,pfq)==0 fprintf(' k=%4d    epsm=%9.3e  \n',k,stopc); end;     %(6)
    x0=x;   Fx0=Fx;    k=k+1;                                            %(7)
    x=max(x0-Fx0*beta,0);        Fx=d.*atan(x) + M*x + q;    l=l+1;      %(8)
       dx=x0-x;    df=(Fx0-Fx)*beta;                                     %(9)
       r=norm(df)/norm(dx);                                              %(10)
       while r>0.9     beta=0.7*beta*min(1,1/r);    l=l+1;               %(11)
         x=max(x0-Fx0*beta,0);     Fx=d.*atan(x) + M*x + q;              %(12)
         dx=x0-x;    df=(Fx0-Fx)*beta;     r=norm(df)/norm(dx);          %(13)
       end;                                                              %(14)
       dxf=dx-df;    r1=dx'*dxf;    r2=dxf'*dxf;     alpha=r1/r2;        %(15)
       x=max(x0- Fx*beta*alpha*1.9,0);                                   %(16)
       Fx= d.*atan(x) + M*x + q;     l=l+1;                              %(17)
       ex=x-max(x-Fx,0);     stopc=norm(ex,inf);                         %(18)
       if r <0.4  beta=beta*1.5;  end;                                   %(19)
end;  toc;   fprintf(' k=%4d   epsm=%9.3e     l=%4d  \n',k,stopc,l);    %%%%
```

NCP 的计算结果 1　Easy Problems $q \in (-500, 500)$

| | PC Method-I | | | PC Method-II | | |
|---|---|---|---|---|---|---|
| $n =$ | No. It | No. F | CPU | No. It | No. F | CPU |
| 500 | 448 | 941 | 0.15 | 372 | 792 | 0.12 |
| 1000 | 475 | 995 | 1.37 | 410 | 852 | 1.17 |
| 1500 | 507 | 1064 | 3.17 | 416 | 887 | 2.64 |
| 2000 | 515 | 1080 | 5.53 | 418 | 892 | 4.55 |

NCP 的计算结果 2　Hard Problems $q \in (-500, 0)$

| | PC Method-I | | | PC Method-II | | |
|---|---|---|---|---|---|---|
| $n =$ | No. It | No. F | CPU | No. It | No. F | CPU |
| 500 | 908 | 1913 | 0.30 | 799 | 1704 | 0.27 |
| 1000 | 980 | 2068 | 2.87 | 857 | 1824 | 2.53 |
| 1500 | 941 | 1983 | 5.88 | 834 | 1771 | 5.25 |
| 2000 | 1112 | 2352 | 12.18 | 986 | 2105 | 10.87 |

PC Method-II converges faster than PC Method-I.

♣ 程序在附件的 Codes-03 中：运行 demo.m 输入 $n$ 就可以, 其中也可以选择不同问题类型. PCd1.m 和 PCd2.m 分别是 PC Method-I 和 PC Method-II 的子程序.



# 5　更一般的孪生方法的统一框架

我们在 [11, 12] 中给出了求解变分不等式更一般的孪生收缩算法.

首先, 定义了预测点(或称检验点) . 对给定的 $u^k$, 根据一定法则生成的 $\tilde{u}^k \in \Omega$ 说成是一个预测点, 如果 $\boxed{u^k = \tilde{u}^k \quad \Leftrightarrow \quad u^k \in \Omega^*.}$

例如, 对给定的 $u^k$ 和 $\beta > 0$, 由投影 $\boxed{\tilde{u}^k = P_\Omega[u^k - \beta F(u^k)]}$ 给出的 $\tilde{u}^k$ 是按确定的法则给出的, 它是一个预测点, 但这不是给出预测点的惟一方法.

**统一框架**. 对给定的 $u^k$, 设 $\tilde{u}^k \in \Omega$ 是 $u^k$ 的一个预测点. 设有基于 $(u^k, \tilde{u}^k)$ 的一对孪生的方向 $d_1(u^k, \tilde{u}^k)$, $d_2(u^k, \tilde{u}^k)$ 和误差度量函数 $\varphi(u^k, \tilde{u}^k) \geq 0$, 它们满足以下条件:

1. 它们满足关系式
$$\tilde{u}^k \in \Omega, \quad (u - \tilde{u}^k)^T d_2(u^k, \tilde{u}^k) \geq (u - \tilde{u}^k)^T d_1(u^k, \tilde{u}^k), \quad \forall u \in \Omega. \tag{5.1a}$$

2. 存在常数 $K > 0$, 使得
$$\|d_1(u^k, \tilde{u}^k)\| \leq K\|u^k - \tilde{u}^k\|. \tag{5.1b}$$

3. 对任意的 $u^* \in \Omega^*$, 有

$$(\tilde{u}^k - u^*)^T d_2(u^k, \tilde{u}^k) \geq \varphi(u^k, \tilde{u}^k) - (u^k - \tilde{u}^k)^T d_1(u^k, \tilde{u}^k), \tag{5.1c}$$

4. $\varphi(u^k, \tilde{u}^k)$ 是 VI$(\Omega, F)$ 的误差度量函数, 即存在常数 $\delta > 0$, 使得

$$\varphi(u^k, \tilde{u}^k) \geq \delta \|u^k - \tilde{u}^k\|^2 \quad \& \quad \varphi(u^k, \tilde{u}^k) = 0 \;\Leftrightarrow\; u^k = \tilde{u}^k. \tag{5.1d}$$

对误差度量函数 $\varphi(u^k, \tilde{u}^k)$ 而言, $d_1(u^k, \tilde{u}^k), d_2(u^k, \tilde{u}^k)$ 都是有利方向.

**Lemma 5.1** 如果统一框架中的条件 (5.1a) 和 (5.1c) 满足, 则有

$$(u^k - u^*)^T d_1(u^k, \tilde{u}^k) \geq \varphi(u^k, \tilde{u}^k), \quad \forall u^k \in \Re^n, u^* \in \Omega^*. \tag{5.2}$$

**证明**. 因为 $u^* \in \Omega$, 以 $u^*$ 代 (5.1a) 中的 $u$, 就有

$$(\tilde{u}^k - u^*)^T d_1(u^k, \tilde{u}^k) \geq (\tilde{u}^k - u^*)^T d_2(u^k, \tilde{u}^k).$$

再根据条件 (5.1c), 得到

$$(\tilde{u}^k - u^*)^T d_1(u^k, \tilde{u}^k) \geq \varphi(u^k, \tilde{u}^k) - (u^k - \tilde{u}^k)^T d_1(u^k, \tilde{u}^k)$$

从上式直接得到 (5.2), 引理得证. $\square$



**Lemma 5.2** 如果统一框架中的条件 (5.1a) 和 (5.1c) 满足, 则有

$$(u^k - u^*)^T d_2(u^k, \tilde{u}^k) \geq \varphi(u, \tilde{u}), \quad \forall u^k \in \Omega, u^* \in \Omega^*. \tag{5.3}$$

**证明**. 因为 $u^k \in \Omega$, 以 $u^k$ 代 (5.1a) 中的 $u$, 我们有

$$(u^k - \tilde{u}^k)^T d_2(u^k, \tilde{u}^k) \geq (u^k - \tilde{u}^k)^T d_1(u^k, \tilde{u}^k). \tag{5.4}$$

将 (5.4) 和 (5.1c) 相加, 得到

$$(u^k - u^*)^T d_2(u^k, \tilde{u}^k) \geq \varphi(u^k, \tilde{u}^k).$$

引理得证. $\square$

根据提供的 $d_1(u^k, \tilde{u}^k)$ 和 $d_2(u^k, \tilde{u}^k)$ 一对孪生方向, 我们可以构造一对算法

(Contraction Method-I) $\quad u^{k+1} = u^k - \gamma \alpha_k^* d_1(u^k, \tilde{u}^k),$

(Contraction Method-II) $\quad u^{k+1} = P_\Omega\{u^k - \gamma \alpha_k^* d_2(u^k, \tilde{u}^k)\},$

其中 $\alpha_k^* = \varphi(u^k, \tilde{u}^k) / \|d_1(u^k, \tilde{u}^k)\|^2, \gamma \in (0, 2)$, 由(5.1b)和(5.1d), 步长是有界的.

孪生方向, 相同步长, 是 PC 方法中最优美的篇章. 证明需技巧, 使用很方便!

后记 无约束优化的梯度类算法中, Barzilai-Borwein 算法 [1] **(BB 算法)** 比最速下降法收敛快得多. 如果用一句话向他人介绍 **BB** 算法, 那就相当于在最速下降法中用当前点的梯度和上一步迭代的最优步长. 根据我们的计算经验, 对无约束凸二次优化用最速下降法能达到的收敛精度, 若改用 **BB** 算法, 收敛速度往往有数量级的提高. **BB** 算法的优秀数值表现已经在工程界得到广泛认可. Dai 和 Yuan [3, 4] 对这类算法做了深入的理论研究. 据此美妙结合, Yuan 还引伸了一个美丽的一般原则, 令人拍手叫绝!

求解单调变分不等式的投影收缩算法中, 什么方法最好? 如果也要用一句话向人介绍, 就是采用的孪生方法中的PC Method-II, 因为投影容易实现. 根据前一讲的内容, 已经找到了一个寻查方向, 事实上还存在另一个孪生方向. 采用这个孪生方向, 还用前一讲提供的同样的步长, 设计的方法往往会比第二讲中精细化后的外梯度方法收敛快一倍以上. 据此也可对应一个朴实的一般原则, 那就是: 兄弟齐心, 其利断金。

孪生方向, 相同步长的证明 [8], 经过了一个相当漫长的过程. 彼时彼景, 至今历历在目。

# References


[1] J. Barzilai and J.M. Borwein, Two-point step size gradien methods. *IMA J. Numerical Analysis* **8** 141–148, 1988

[2] E. Blum and W. Oettli, Mathematische Optimierung. Grundlagen und Verfahren. Ökonometrie und Unternehmensforschung. Berlin-Heidelberg-New York: Springer-Verlag, 1975.

[3] Y.H. Dai and L.Z. Liao, R-linear convergence of the Barzilai and Borwein gradient method. *IMA J. Numerical Analysis* **22**, 1–10, 2002.

[4] Y.H. Dai and Y. Yuan, Alternate minimization gradient method. *IMA J. Numerical Analysis* **23**, 377–393, 2003.





[5] D. S. Guo and Y.N. Zhang, Simulation and experimental verification of weighted velocity and acceleration minimization for robotic redundancy resolution, *IEEE Transactions on Automation Science and Engineering*, 2014, 11: 1203–1217.

[6] B.S. He, A new method for a class of linear variational inequalities, *Math. Progr.*, **66**, 137–144, 1994.

[7] B.S. He, Solving a class of linear projection equations, *Numerische Mathematik*, **68**, 71–80, 1994.

[8] B.S. He, A globally linearly convergent projection and contraction method for a class of linear complementarity problems. Schwerpunktprogramm der Deutschen Forschungsgemeinschaft Anwendungsbezogene Optimierung und Steuerung, Report No. 352, 1992

[9] B.S. He, A class of projection and contraction methods for monotone variational inequalities, *Applied Mathematics and optimization*, bf 35, 69–76, 1997.

[10] B.S He and L.-Z Liao, Improvements of some projection methods for monotone nonlinear variational inequalities, JOTA, **112**, 111-128, 2002

[11] B.S. He, L.Z. Liao, and X. Wang, Proximal-like contraction methods for monotone variational inequalitiesin a unified framework I: Effective quadruplet and primary methods, Comput. Optim. Appl., **51**, 649-679, 2012.

[12] B.S. He, L.Z. Liao, and X. Wang, Proximal-like contraction methods for monotone variational inequalities in a unified framework II: General methods and numerical experiments, Comput. Optim. Appl. **51**, 681-708, 2012

[13] B. S. He and M.-H. Xu, A general framework of contraction methods for monotone variational inequalities, Pacific J. Optimization, **4**, 195-212, 2008.

[14] B.S. He, X.M. Yuan and J.J.Z. Zhang, *Comparison of two kinds of prediction-correction methods for monotone variational inequalities*, Computational Optimization and Applications, **27**, 247-267, 2004.

[15] D.F. Sun, A class of iterative methods for solving nonlinear projection equations, JOTA **91**, 123-140, 1996.

[16] L. Xiao and Y. N. Zhang, Acceleration-level repetitive motion planning and its experimental verification on six-link planar robot manipulator, *IEEE Transactions on Control System Technology*, 2013, 21: 906–914.

[17] H. Zheng, F. Liu and X.L. Du, Complementarity problem arising from static growth of multiple cracks and MLS-based numerical manifold method, Computer Methods in Applied Mechanics and Engineering, 295 (2015) 150-171.

[18] H. Zheng, P. Zhang and X.L. Du, Dual form of discontinuous deformation analysis, Computer Methods in Applied Mechanics and Engineering, 305 (2016) 196-216.

# 凸优化和单调变分不等式的收缩算法

## 第四讲：为线性约束凸优化 定制的 PPA 算法及其应用

Customized PPA for linearly constrained Optimization and its applications

南京大学数学系　　何炳生
hebma@nju.edu.cn

The context of this lecture is based on the publications [6, 7]



## 线性约束的凸优化问题

从这一讲开始的四讲, 讨论的问题是

$$\boxed{\min\{\theta(x) \mid Ax = b\,(or\ Ax \ge b)\ x \in \mathcal{X}\}}$$

其中 $\theta(x)$ 是凸函数, $A \in \Re^{m\times n}$, $b \in \Re^m$, $\mathcal{X}$ 是 $\Re^n$ 中的闭凸集.

对任意给定的 $r > 0$ 和 $a \in \Re^n$, 通篇我们假设子问题

$$\boxed{\min\{\theta(x) + \frac{r}{2}\|x - a\|^2 \mid x \in \mathcal{X}\}}$$

的求解是简单的.

**目的**　说明将线性约束的凸优化问题转换成混合单调变分不等式, 选择适当的正定矩阵 $G$, 采用 $G$-模下的 PPA 方法. 在上述假设条件下, 线性约束的凸优化问题就变得非常容易求解.

# 1 PPA 算法的基本性质

## 1.1 简单约束凸优化问题的PPA 算法

简单约束凸优化问题是指 ($\theta(x)$ 也可以是非光滑的)

$$\min\{\theta(x) \mid x \in \mathcal{X}\}. \tag{1.1}$$

用邻近点算法 (**Proximal Point Algorithm**) [9, 11] 求解的基本步骤:

对给定的 $r > 0$ 和 $x^k$, 令

$$\tilde{x}^k = \text{Argmin}\{\theta(x) + \frac{r}{2}\|x - x^k\|^2 \mid x \in \mathcal{X}\}. \tag{1.2}$$

如果 $x^k \in \mathcal{X}$, 根据 (1.2), 我们有

$$\theta(\tilde{x}^k) + \frac{r}{2}\|\tilde{x}^k - x^k\|^2 \leq \theta(x^k).$$

也就是说, 函数值 $\theta(\tilde{x}^k)$ 比 $\theta(x^k)$ 小了至少 $\frac{r}{2}\|\tilde{x}^k - x^k\|^2$.

我们考虑在收缩算法的意义下如何让新的迭代点更靠近解集. 根据凸优化问



题 (1.2) 的最优性条件, 有

$$\tilde{x}^k \in \mathcal{X}, \quad (x - \tilde{x}^k)^T\{\partial\theta(\tilde{x}^k) + r(\tilde{x}^k - x^k)\} \geq 0, \quad \forall\, x \in \mathcal{X}, \tag{1.3}$$

其中 $\partial\theta(x)$ 是 $\theta(x)$ 的次梯度. 对凸函数, 我们有以下的基本性质:

$$\tilde{x}^k \in \mathcal{X}, \quad \theta(x) - \theta(\tilde{x}^k) \geq (\partial\theta(\tilde{x}^k))^T(x - \tilde{x}^k), \quad \forall\, x \in \mathcal{X}.$$

以此代入 (1.3), 就有

$$\tilde{x}^k \in \mathcal{X}, \ \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T\{r(\tilde{x}^k - x^k)\} \geq 0, \ \forall\, x \in \mathcal{X}.$$

由于 $x^* \in \mathcal{X}$, 以 $x^*$ 代上式中的 $x$, 得到

$$(\tilde{x}^k - x^*)^T(x^k - \tilde{x}^k) \geq \frac{1}{r}\big(\theta(\tilde{x}^k) - \theta(x^*)\big). \tag{1.4}$$

因为 $\tilde{x}^k \in \mathcal{X}$, (1.4) 的右端非负. 由此可以得到

$$\{(x^k - x^*) - (x^k - \tilde{x}^k)\}^T(x^k - \tilde{x}^k) \geq 0,$$

进而

$$(x^k - x^*)^T(x^k - \tilde{x}^k) \geq \|x^k - \tilde{x}^k\|^2. \tag{1.5}$$

我们取

$$x^{k+1}(\alpha) = x^k - \alpha(x^k - \tilde{x}^k), \quad \alpha \in (0,2) \tag{1.6}$$

为新的迭代点, 利用 (1.5) 就有

$$\begin{aligned}
\vartheta(\alpha) \quad &:= \quad \|x^k - x^*\|^2 - \|x^{k+1}(\alpha) - x^*\|^2 \\
&= \quad \|x^k - x^*\|^2 - \|(x^k - x^*) - \alpha(x^k - \tilde{x}^k)\|^2 \\
&= \quad 2\alpha(x^k - x^*)^T(x^k - \tilde{x}^k) - \alpha^2\|x^k - \tilde{x}^k\|^2 \\
&\geq \quad 2\alpha\|x^k - \tilde{x}^k\|^2 - \alpha^2\|x^k - \tilde{x}^k\|^2 =: q(\alpha).
\end{aligned}$$

$q(\alpha)$ 是 $\alpha$ 的二次函数, 它在 $\alpha = 1$ 时候取得极大值. 如同在上一讲中同样的原因, 我们取 $\alpha \in [1.2, 2)$, 往往会加快收敛速度.

**PPA 算法对简单约束凸优化问题收敛性质**

由 PPA 公式 (1.6) 产生的迭代序列 $\{x^k\}$ 满足收缩性质

$$\|x^{k+1} - x^*\|^2 \leq \|x^k - x^*\|^2 - \alpha(2-\alpha)\|x^k - \tilde{x}^k\|^2.$$

我们称这样的序列 $\{x^k\}$ 为 Fejér 单调的.



## 1.2 欧氏模下混合单调变分不等式的 PPA 算法

我们考虑如下的混合单调变分不等式

$$\text{VI}(\Omega, F, \theta) \quad u^* \in \Omega, \quad \theta(x) - \theta(x^*) + (u - u^*)^T F(u^*) \geq 0, \quad \forall\, u \in \Omega, \tag{1.7}$$

其中 $x$ 是 $u$ 的子向量. $\theta(\cdot)$ 是 $x$ 的凸函数(不一定可微), $F$ 是 $u$ 的单调算子.
用邻近点算法求解混合单调变分不等式 (1.7), 对给定的 $u^k \in \Re^n$ 和 $r > 0$,
$k$-次迭代中的子问题是

$$\text{(PPA)} \qquad u \in \Omega, \quad \theta(x') - \theta(x) + (u' - u)^T F_k(u) \geq 0, \quad \forall\, u' \in \Omega, \tag{1.8a}$$

其中

$$F_k(u) = F(u) + r(u - u^k). \tag{1.8b}$$

设 $\tilde{u}^k$ 是 (1.8) 的解, 那么, 我们有

$$\tilde{u}^k \in \Omega, \;\; \theta(x') - \theta(\tilde{x}^k) + (u' - \tilde{u}^k)^T \big(F(\tilde{u}^k) + r(\tilde{u}^k - u^k)\big) \geq 0, \; \forall u' \in \Omega, \tag{1.9}$$

以 $u^*$ 替代上式中的 $u'$, 得到

$$(\tilde{u}^k - u^*)^T(u^k - \tilde{u}^k) \geq \frac{1}{r}\{(\tilde{u}^k - u^*)^T F(\tilde{u}^k) + \theta(\tilde{x}^k) - \theta(x^*)\}. \tag{1.10}$$

由于 $F$ 是单调算子, 因此有

$$(\tilde{u}^k - u^*)^T F(\tilde{u}^k) \geq (\tilde{u}^k - u^*)^T F(u^*).$$

因为 $\tilde{u}^k \in \Omega$ 并且 $u^*$ 是 VI$(\Omega, F)$ 的解, 根据混合变分不等式 (1.7) 的定义, 有

$$\theta(\tilde{x}^k) - \theta(x^*) + (\tilde{u}^k - u^*)^T F(u^*) \geq 0.$$

根据 (1.10) 和上面两个关系式得到下面的关键不等式

$$(u^k - u^*)^T (u^k - \tilde{u}^k) \geq \|u^k - \tilde{u}^k\|^2, \quad \forall\, u^* \in \Omega^*. \tag{1.11}$$

我们可以取 $\gamma \in [1, 2)$, 用公式

$$u^{k+1} = u^k - \gamma(u^k - \tilde{u}^k) \tag{1.12}$$

产生下一个迭代点. 用不等式 (1.11) 可以推出, 序列 $\{u^k\}$ 满足关系式

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \gamma(2-\gamma)\|u^k - \tilde{u}^k\|^2.$$

我们说, 对解集 $\Omega^*$, 序列$\{u^k\}$ 是Fej$\acute{e}$r 单调的.



## 1.3 $G$-模下混合单调变分不等式的 PPA 算法

假如我们将 (1.8) 中的常数 $r > 0$ 换成一个对称正定矩阵 $G$, (1.9) 就变成

$$\tilde{u}^k \in \Omega, \;\; \theta(x') - \theta(\tilde{x}^k) + (u' - \tilde{u}^k)^T \big(F(\tilde{u}^k) + G(\tilde{u}^k - u^k)\big) \geq 0, \;\; \forall u' \in \Omega. \tag{1.13}$$

将 (1.13) 中的 $u'$ 置换成 $u^*$ 并利用 $F$ 的单调性, 就有

$$(\tilde{u}^k - u^*)^T G(u^k - \tilde{u}^k) \geq \theta(\tilde{x}^k) - \theta(x^*) + (\tilde{u}^k - u^*)^T F(u^*), \quad \forall\, u^* \in \Omega.$$

由于上式右端非负, 由此导出

$$(u^k - u^*)^T G(u^k - \tilde{u}^k) \geq \|u^k - \tilde{u}^k\|_G^2, \quad \forall\, u^* \in \Omega^*. \tag{1.14}$$

假设新的迭代点

$$u^{k+1}(\alpha) = u^k - \alpha(u^k - \tilde{u}^k), \tag{1.15}$$

在收缩算法的框架下考察与 $\alpha$ 相关的 $G$-模下距离平方缩短量

$$\vartheta(\alpha) = \|u^k - u^*\|_G^2 - \|u^{k+1}(\alpha) - u^*\|_G^2. \tag{1.16}$$

利用 (1.14) 就有

$$\begin{aligned}\vartheta(\alpha) &= \|u^k - u^*\|_G^2 - \|u^k - u^* - \alpha(u^k - \tilde{u}^k)\|_G^2 \\ &\geq 2\alpha\|u^k - \tilde{u}^k\|_G^2 - \alpha^2\|u^k - \tilde{u}^k\|_G^2. \end{aligned} \tag{1.17}$$

我们得到 $\vartheta(\alpha)$ 的一个下界、二次函数 $q(\alpha)$,

$$q(\alpha) = (2\alpha - \alpha^2)\|u^k - \tilde{u}^k\|_G^2. \tag{1.18}$$

使 $q(\alpha)$ 达到极大值的 $\alpha^* = 1$. 注意到收缩算法的本意是要在 $G$-模下极大化 $\vartheta(\alpha)$ (见 (1.16)), 由于它含有未知的 $u^*$, 我们不得已才极大化它的下界函数 $q(\alpha)$. 因此, 在实际计算中, 我们令

$$u^{k+1} = u^k - \gamma(u^k - \tilde{u}^k), \quad \gamma \in (1,2) \tag{1.19}$$

其中 $\gamma \in (1,2)$ 称为松弛因子. 由 **PPA** 产生的序列 $\{u^k\}$ 是Fejér 单调的, 即有

$$\|u^{k+1} - u^*\|_G^2 \leq \|u^k - u^*\|_G^2 - \gamma(2-\gamma)\|u^k - \tilde{u}^k\|_G^2. \tag{1.20}$$

我们称 $\gamma = 1$ 的PPA 算法为经典的 (Classical) PPA 算法, 对 $\gamma \in (1,2)$ 的PPA 算法为外延的 (Extended) PPA 算法.

**Theorem 1.1** *对给定的 $u^k$, 设 $\tilde{u}^k$ 由* (1.13) *生成且新的迭代点 $u^{k+1}$ 由*



(1.19) *给出. 那么迭代序列 $\{u^k\}$ 收敛于变分不等式* (1.7) *的一个解点.*

证明要点:

- 因为 $G$ 是正定矩阵, 由于 (1.20) 对 (1.7) 的每个解点都成立, 我们得到序列 $\{u^k\}$ 有界和 $\lim_{k\to\infty}\|u^k - \tilde{u}^k\|_G = 0$ 这样两个基本事实.
- 由于 $\lim_{k\to\infty}\|u^k - \tilde{u}^k\|_G = 0$ 和 $\{u^k\}$ 有界, $\{\tilde{u}^k\}$ 也有界. 同时, $\{u^k\}$ 和 $\{\tilde{u}^k\}$ 有同样的聚点.
- 根据 (1.13), $\{\tilde{u}^k\}$ 的每个聚点（同时也是 $\{u^k\}$ 的聚点）都是变分不等式的解点.
- 最后由不等式 (1.20) 说明这个方法生成的序列 $\{u^k\}$ 只能有一个聚点从而方法是总体收敛的.

这一讲要在变分不等式框架下, 用 PPA 方法求解线性约束凸优化.

## 2 线性约束凸优化等价的单调变分不等式

线性约束的凸优化问题

$$\min\{\theta(x) \mid Ax = b,\ x \in \mathcal{X}\} \tag{2.1}$$

的 Lagrange 函数是定义在 $\mathcal{X} \times \Re^m$ 上的

$$L(x,\lambda) = \theta(x) - \lambda^T(Ax - b).$$

设 $(x^*, \lambda^*)$ 是 Lagrange 函数的一个鞍点, 便有

$$L_{\lambda\in\Re^m}(x^*,\lambda) \le L(x^*,\lambda^*) \le L_{x\in\mathcal{X}}(x,\lambda^*)$$

因此, 求 Lagrange 函数的一个鞍点等价于求 $(x^*, \lambda^*)$ 使其满足:

$$\left\{\begin{array}{ll} x^* \in \mathcal{X}, & \theta(x) - \theta(x^*) + (x - x^*)^T(-A^T\lambda^*) \ge 0, \quad \forall x \in \mathcal{X}, \\ \lambda^* \in \Re^m, & Ax^* - b = 0. \end{array}\right. \tag{2.2}$$



通过定义

$$u = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F(u) = \begin{pmatrix} -A^T\lambda \\ Ax - b \end{pmatrix} \tag{2.3}$$

和

$$\Omega = \mathcal{X} \times \Re^m, \tag{2.4}$$

求解 (2.1) 相当于求解混合变分不等式 (1.7). 注意到 $\theta(x)$ 是凸函数, (2.3) 中的算子 $F$ 是仿射单调的.

**如果我们处理的线性约束的凸优化问题是**

$$\min\{\theta(x) \mid Ax \ge b,\ x \in \mathcal{X}\}. \tag{2.5}$$

它的 Lagrange 函数是定义在 $\mathcal{X} \times \Re^m_+$ 上的

$$L(x,\lambda) = \theta(x) - \lambda^T(Ax - b). \tag{2.6}$$

在 $\mathcal{X} \times \Re^m_+$ 上求 $L(x,\lambda)$ 的一个鞍点等价于求 $(x^*, \lambda^*)$:

$$\left\{\begin{array}{lll} x^* \in \mathcal{X}, & \theta(x) - \theta(x^*) + (x - x^*)^T(-A^T\lambda^*) \ge 0, & \forall x \in \mathcal{X}, \\ \lambda^* \ge 0, & (\lambda - \lambda^*)^T(Ax^* - b) \ge 0, & \forall \lambda \ge 0. \end{array}\right. \tag{2.7}$$

这仍然是一个形如 (1.7) 的混合变分不等式

$$u^* \in \Omega, \quad \theta(x) - \theta(x^*) + (u - u^*)^T F(u^*) \geq 0, \quad \forall\, u \in \Omega. \tag{2.8}$$

其中的算子 $F$ 与 (2.3) 中的完全相同, 与 (2.1) 问题不同的只是

$$\Omega = \mathcal{X} \times \Re^m_+.$$

**我们通过求解相应的混合变分不等式来求解线性约束的凸优化问题.**

以后我们记 $\Omega = \mathcal{X} \times \Lambda$. 对问题 (2.1), $\Lambda = \Re^m$; 对问题 (2.5), $\Lambda = \Re^m_+$.

**Customized PPA 框架:** 对给定的 $u^k$, 我们生成 $\tilde{u}^k \in \Omega$ 使得它是

$$\tilde{u}^k \in \Omega, \ \theta(x) - \theta(\tilde{x}^k) + (u - \tilde{u}^k)^T \{F(\tilde{u}^k) + G(\tilde{u}^k - u^k)\} \geq 0, \ \forall\, u \in \Omega, \tag{2.9}$$

其中 $G$ 是我们考虑 $G$-模下的收缩算法而刻意找的对称正定矩阵. 以

$$u^{k+1} = u^k - \gamma(u^k - \tilde{u}^k), \quad \gamma \in (0, 2)$$

生成新的迭代点. 由于这样的 $G$ 是根据需要构造的, 因而称为 Customized PPA.



# 3 变分不等式框架下凸优化的PPA 算法

对等式约束凸优化问题 (2.1) 和不等式约束凸优化问题 (2.5),

$$\boxed{\min\{\theta(x) \mid Ax = b \, (or \ Ax \geq b), \ x \in \mathcal{X}\}}$$

用 Customized PPA 求解与它们等价的变分不等式 (2.2) (或者(2.7)). 要求 $A^TA$ 的模是容易估计的. 这一讲的最后一节会说明一些有应用背景的问题, 恰能满足这些要求.

## 3.1 Primal-dual hybrid gradient algorithm

Since the objective is to find a saddle point of the Lagrange function, a natural idea is to use the primal-dual hybrid gradient algorithm [12]. For given $(x^k, \lambda^k)$, by using the primal-dual hybrid gradient algorithm,

$$x^{k+1} = \text{Argmin}\{L(x, \lambda^k) + \frac{r}{2}\|x - x^k\|^2 \,|\, x \in \mathcal{X}\}, \tag{3.1}$$

after getting $x^{k+1}$, we obtain

$$\lambda^{k+1} = \text{Argmax}\{L(x^{k+1},\lambda) - \frac{s}{2}\|\lambda - \lambda^k\|^2 \,|\, \lambda \in \Re^m\}. \tag{3.2}$$

Combining (3.1) and (3.2), we get

$$\theta(x) - \theta(x^{k+1}) + \begin{pmatrix} x - x^{k+1} \\ \lambda - \lambda^{k+1} \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\lambda^{k+1} \\ Ax^{k+1} - b \end{pmatrix} + \begin{pmatrix} r(x^{k+1} - x^k) + A^T(\lambda^{k+1} - \lambda^k) \\ s(\lambda^{k+1} - \lambda^k) \end{pmatrix} \right\} \ge 0, \quad \forall (x,\lambda) \in \Omega.$$

The compact form is

$$\theta(x) - \theta(x^{k+1}) + (u - u^{k+1})^T\{F(u^{k+1}) + M(u^{k+1} - u^k)\} \ge 0, \ \forall u \in \Omega, \tag{3.3}$$

where

$$M = \begin{pmatrix} rI_n & A^T \\ 0 & sI_m \end{pmatrix} \qquad \text{is not symmetric.}$$



**Remark** For general, min-max problem, the primal-dual hybrid gradient algorithm is not convergent. For example, the primal-dual linear programming:

$$\text{(Primal)} \quad \begin{array}{ll} \min & x \\ & x = 1 \\ & x \ge 0. \end{array} \qquad \text{(Dual)} \quad \begin{array}{ll} \max & y \\ \text{s. t.} & y \le 1 \end{array}$$

The Lagrange function is

$$L(x,y) = x - y(x-1),$$

which is defined on $R_+ \times R$. The saddle point of the Lagrange function is

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Using the Primal-dual hybrid gradient method with $r = s = 1$. For given $(x^k, y^k)$, Zhu and Chan's procedure is

$$x^{k+1} = \text{Argmin}\{L(x,y^k) + \frac{1}{2}\|x - x^k\|^2 \,|\, x \ge 0\} = \max\{(x^k + y^k - 1), 0\},$$

and then

$$y^{k+1} = \text{Argmax}\{L(x^{k+1}, y) - \frac{1}{2}\|y - y^k\|^2\} = y^k - (x^{k+1} - 1).$$

In other words, the iteration formula is

$$\begin{cases} x^{k+1} = \max\{(x^k + y^k - 1), 0\}, \\ y^{k+1} = y^k - (x^{k+1} - 1). \end{cases}$$

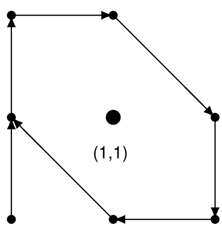
Begin with $(x^0, y^0) = (0, 0)$, then we have

$$\begin{pmatrix} x^0 \\ y^0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \quad \cdots, \quad \begin{pmatrix} x^7 \\ y^7 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Note that we have

$$\begin{pmatrix} x^7 \\ y^7 \end{pmatrix} = \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} x^{k+6} \\ y^{k+6} \end{pmatrix} = \begin{pmatrix} x^k \\ y^k \end{pmatrix}, \ \forall\, k \geq 1.$$






## 3.2 Customized PPA

If we can change $M$ to a symmetric matrix $G$ such that

$$M = \begin{pmatrix} rI_n & A^T \\ 0 & sI_m \end{pmatrix} \quad \Rightarrow \quad G = \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix},$$

then $\tilde{u}^k$ is the solution of the following variational inequality:

$$\tilde{u}^k \in \Omega, \quad \theta(u) - \theta(\tilde{u}^k) + (u - \tilde{u}^k)^T\{F(\tilde{u}^k) + G(\tilde{u}^k - u^k)\} \geq 0, \ \forall u \in \Omega. \quad (3.4)$$

For this purpose, we need only change $(A\tilde{x}^k - b) + s(\tilde{\lambda}^k - \lambda^k) = 0$ to

$$(A\tilde{x}^k - b) + A(\tilde{x}^k - x^k) + s(\tilde{\lambda}^k - \lambda^k) = 0. \tag{3.5}$$

Because $\tilde{x}^k$ is known, with the given $x^k$ and $\lambda^k$, $\tilde{\lambda}^k$ is given by

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}[A(2\tilde{x}^k - x^k) - b].$$

Thus, for given $(x^k, \lambda^k)$, produce a proximal point $(\tilde{x}^k, \tilde{\lambda}^k)$ via (3.1) and (3.5) can be summarized as:

$$\boxed{\begin{aligned} \tilde{x}^k &= \text{argmin}\big\{L(x, \lambda^k) + \frac{r}{2}\|x - x^k\|^2 \,\big|\, x \in \mathcal{X}\big\}. && \text{(3.6a)} \\ \tilde{\lambda}^k &= \text{argmax}\big\{L\big([2\tilde{x}^k - x^k], \lambda\big) - \frac{s}{2}\|\lambda - \lambda^k\|^2\big\} && \text{(3.6b)} \end{aligned}}$$

We call the point $(\tilde{x}^k, \tilde{\lambda}^k)$ generated by (3.6) as the predictor. The subproblem (3.6a) is equivalent to

$$\tilde{x}^k = \text{argmin}\big\{\theta(x) + \frac{r}{2}\big\|x - \big[x^k + \frac{1}{r}A^T\lambda^k\big]\big\|^2 \,\big|\, x \in \mathcal{X}\big\}.$$



The solution of (3.6b) is given by

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}[A(2\tilde{x}^k - x^k) - b].$$

**Indeed, under the assumption, the sub-problem** (3.6a) **is simple.**

In the case that $rs > \|A^TA\|$, the matrix

$$G = \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix} \quad \text{is positive definite}$$

and the $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k)$ generated by (3.6) is a proximal point. Based on the predictor, the new iterate is given by

$$\boxed{u^{k+1} := u^k - \gamma(u^k - u^{k+1}), \quad \gamma \in (0, 2).}$$

**Theorem 3.1** *The sequence* $\{u^k = (x^k, \lambda^k)\}$ *generated by the customized PPA* *satisfies*

$$\|u^{k+1}-u^*\|_G^2 \le \|u^k-u^*\|_G^2 - \gamma(2-\gamma)\|u^k-\tilde{u}^k\|_G^2. \tag{3.7}$$

定理 3.1 中的 (3.7) 是方法总体收敛的关键不等式. 如同 §1.3，可以由此证明算法的总体收敛性. 计算中实际困难往往在于如何选择参数 $r$ 和 $s$.

我们也可以用另外的顺序产生预测点 $\tilde{u}^k=(\tilde{x}^k,\tilde{\lambda}^k)$. 对给定的 $(x^k,\lambda^k)$, 求 $(\tilde{x}^k,\tilde{\lambda}^k)\in\Omega$, 使得

$$\theta(x)-\theta(\tilde{x}^k)+\begin{pmatrix} x-\tilde{x}^k \\ \lambda-\tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k-b \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k-x^k)-A^T(\tilde{\lambda}^k-\lambda^k) \\ -A(\tilde{x}^k-x^k)+s(\tilde{\lambda}^k-\lambda^k) \end{pmatrix} \right\} \ge 0, \quad \forall (x,\lambda)\in\Omega. \tag{3.8}$$

这相当于求得混合变分不等式 (2.2) (或者(2.7)) 的 PPA 子问题 (1.13) 的解, 其中对称矩阵

$$G=\begin{pmatrix} rI_n & -A^T \\ -A & sI_m \end{pmatrix}. \tag{3.9}$$



同样, 当 $rs>\|A^TA\|$ 时 $G$ 是正定的. 注意到(3.8) 的下半部分相当于

$$\tilde{\lambda}^k\in\Lambda, \quad (\lambda-\tilde{\lambda}^k)^T\{(Ax^k-b)+s(\tilde{\lambda}^k-\lambda^k)\}\ge 0, \quad \forall\,\lambda\in\Lambda,$$

它可以由

$$\tilde{\lambda}^k=P_\Lambda[\lambda^k-\frac{1}{s}(Ax^k-b)] \tag{3.10}$$

给出. 因为有了 $\tilde{\lambda}^k$, 变分不等式 (3.8) 的上半部分中未知的只有 $\tilde{x}^k$, 它可以通过求解极小化问题

$$\min\left\{\theta(x)+\frac{r}{2}\Big\|x-\big[x^k+\frac{1}{r}A^T(2\tilde{\lambda}^k-\lambda^k)\big]\Big\|^2 \,\Big|\, x\in\mathcal{X}\right\} \tag{3.11}$$

得到. 根据假设, 这是一个容易求解的问题. 生成预测点以后, 给出新迭代点 $u^{k+1}$ 的公式仍然是

$$u^{k+1}=u^k-\gamma(u^k-\tilde{u}^k), \quad \gamma\in(0,2).$$

# 4 最优化问题中的应用

## 4.1 相关性矩阵校正中的应用

在统计学中, 一个对角元均为 $1$ 的对称半正定矩阵称为 (Correlation Matrix) 相关性矩阵. 对给定的对称矩阵 $C$, 求 $F$-模下与 $C$ 距离最近的相关性矩阵, 其数学表达式是

$$\min\{\frac{1}{2}\|X-C\|_F^2 \mid \mathrm{diag}(X)=e,\ X\in S_+^n\}, \tag{4.1}$$

其中 $e$ 表示每个分量都为 $1$ 的 $n$-维向量, $S_+^n$ 表示 $n\times n$ 正半定锥的集合. 问题 (4.1) 是形如 (2.1) 的等式约束凸优化问题, 其中 $\|A^TA\|=1$.

我们用 $z\in\Re^n$ 作为等式约束 $\mathrm{diag}(X)=e$ 的 Lagrange 乘子.

**Dual-Primal Customized PPA 求解问题**(4.1)

对给定的 $(X^k,z^k)$, 用 (3.10)–(3.11) 产生$(\tilde X^k,\tilde z^k)$:

1. Producing $\tilde z^k$ by
$$\tilde z^k = z^k - \frac{1}{s}(\mathrm{diag}(X^k)-e).$$



2. Finding $\tilde X^k$ which is the solution of the following minimization problem
$$\min\{\frac{1}{2}\|X-C\|_F^2+\frac{r}{2}\|X-[X^k+\frac{1}{r}\mathrm{diag}(2\tilde z^k-z^k)]\|_F^2|X\in S_+^n\}. \tag{4.2}$$

**子问题 (4.2) 求解的具体做法:** 化为等价问题

$$\min\{\frac{1}{2}\|X-\frac{1}{1+r}[rX^k+\mathrm{diag}(2\tilde z^k-z^k)+C]\|_F^2|X\in S_+^n\}.$$

记 $A=\frac{1}{1+r}[rX^k+\mathrm{diag}(2\tilde z^k-z^k)+C]$, 我们只要考虑如何求解

$$\tilde X^k = \text{Argmin}\{\frac{1}{2}\|X-A\|_F^2 \mid X\in S_+^n\}. \tag{4.3}$$

实际上, 将对称矩阵 $A$ 做标准特征值– 特征向量分解

$$A=V\Lambda V^T, \qquad \Lambda=\mathrm{diag}(\lambda_1,\lambda_2,\ldots,\lambda_n). \tag{4.4}$$

其中 $V$ 是正交矩阵. 注意到正交变换下矩阵的 Frobenius-模是不变的, 我们有

$$\|X-A\|_F=\|X-V\Lambda V^T\|_F=\|V^TXV-\Lambda\|_F.$$

要使 $\tilde{X} \succeq 0$ 并且上式右端最小, 应该有

$$V^T \tilde{X} V = \tilde{\Lambda},$$

其中

$$\tilde{\Lambda} = \mathrm{diag}(\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_n), \qquad \tilde{\lambda}_j = \max\{0, \lambda_j\}.$$

最后通过

$$\tilde{X} = V \tilde{\Lambda} V^T$$

得到 $\tilde{X}$. 因此, 每次迭代的主要工作是做 (4.4) 中的特征值 (特征向量) 分解.

**数值试验**　为生成试验例子, 只要给定对称矩阵 $C$.

C=rand(n,n); C=(C'+C)-ones(n,n) + eye(n)

这样的矩阵 $C$ 的对角元在 $(0,2)$ 之间, 非对角元在 $(-1,1)$ 之间。

### Code 4.a. Matlab code for Creating the test examples

```
clear;  close all;
n = 1000;    tol=1e-5;     r=2.0;     s=1.01/r;     gamma=1.5;
rand('state',0);   C=rand(n,n);    C=(C'+C)-ones(n,n) + eye(n);
```



### Code 4.1. Matlab code of the classical PPA

```
%%%     Classical PPA for calibrating correlation matrix          %(1)
function PPAC(n,C,r,s,tol)                                         %(2)
X=eye(n);      y=zeros(n,1);    tic;      %%  The initial iterate %(3)
stopc=1;                          k=0;                             %(4)
while (stopc>tol && k<=100)          %% Beginning of an Iteration  %(5)
 if mod(k,20)==0 fprintf(' k=%4d   epsm=%9.3e  \n',k,stopc); end;  %(6)
   X0=X;     y0=y;     k=k+1;                                      %(7)
   yt=y0 - (diag(X0)-ones(n,1))/s;                 EY=y0-yt;       %(8)
   A=(X0*r + C + diag(yt*2-y0))/(1+r);                             %(9)
   [V,D]=eig(A);   D=max(0,D);  XT=(V*D)*V';     EX=X0-XT;        %(10)
   ex=max(max(abs(EX)));  ey=max(abs(EY));   stopc=max(ex,ey);    %(11)
   X=XT;                  y=yt;                                   %(12)
end;                                         % End of an Iteration %(13)
toc;                      TB = max(abs(diag(X-eye(n))));          %(14)
fprintf(' k=%4d  epsm=%9.3e   max|X_jj - 1|=%8.5f \n',k,stopc,TB); %%
```

做 (4.4) 中的特征值 (特征向量) 分解, 在上述程序中的第 (10) 行用 Matlab 中的语句 [V,D]=eig(A) 实现的, 这是一个计算量大概 $9n^3$ 的运算.

将 Classical PPA 改成 Extended PPA, 只要将第 (12) 行改一下。

## Code 4.2 Matlab Code of the Extended PPA

```
%%%     Extended PPA for calibrating correlation matrix              %(1)
function PPAE(n,C,r,s,tol,gamma)                                     %(2)
X=eye(n);       y=zeros(n,1);     tic;      %%  The initial iterate  %(3)
stopc=1;                        k=0;                                 %(4)
while (stopc>tol && k<=100)          %% Beginning of an Iteration  %(5)
 if mod(k,20)==0 fprintf(' k=%4d   epsm=%9.3e  \n',k,stopc); end;  %(6)
    X0=X;     y0=y;     k=k+1;                                       %(7)
    yt=y0 - (diag(X0)-ones(n,1))/s;               EY=y0-yt;          %(8)
    A=(X0*r + C + diag(yt*2-y0))/(1+r);                              %(9)
    [V,D]=eig(A);    D=max(0,D);  XT=(V*D)*V';     EX=X0-XT;        %(10)
    ex=max(max(abs(EX)));  ey=max(abs(EY));   stopc=max(ex,ey);     %(11)
    X=X0-EX*gamma;          y=y0-EY*gamma;                          %(12)
end;                                           % End of an Iteration %(13)
toc;                        TB = max(abs(diag(X-eye(n))));          %(14)
fprintf(' k=%4d  epsm=%9.3e   max|X_jj - 1|=%8.5f \n',k,stopc,TB); %%
```

两个不同的方法, 程序都很简单, 用不了几行. 两个程序不同的地方仅仅是第 12 行有些差别, 取 $\gamma = 1.5$ 的方法效果却有明显的提高。

由于在这个问题中, $A$ 是投影矩阵, $\|A^T A\| = 1$, 我们只需要选 $rs > 1$ 就能确保 $G \succ 0$. 注意到, 为了平衡 primal 和 dual 残量, 我们取 $r = 2$, $s = 1.01/r$.



特征值分解用 Matlab 中的 eig 的迭代次数和计算时间

| $n \times n$ Matrix | Classical PPA | | Extended PPA | |
|---|---|---|---|---|
| $n =$ | No. It | CPU Sec. | No. It | CPU Sec. |
| 100 | 31 | 0.35 | 23 | 0.26 |
| 200 | 34 | 2.00 | 25 | 1.48 |
| 500 | 39 | 17.28 | 27 | 13.14 |
| 800 | 41 | 72.82 | 29 | 51.97 |
| 1000 | 47 | 153.14 | 31 | 102.12 |
| 2000 | 62 | 1344.50 | 40 | 881.38 |

The extended PPA converges faster than the classical PPA.

$$\frac{\text{It. No. of Extended PPA}}{\text{It. No. of Classical PPA}} \approx 65\%.$$

♣ 关于相关系数矩阵校正的程序在附件的 Codes-04 的文件夹“*矩阵校正*”中. 只要运行 demo.m, 输入 n 就可以了. 其中的 PPAC.m 和 PPAE.m 分别是 Classical PPA 和 Extended PPA 的子程序.

确实, 用这一讲介绍的 PPA 方法求解相关矩阵矩阵校正问题, 每步迭代的主要计算工作量是对一个对称矩阵用 Matlab 中的标准子程序做 [V,D]=eig(A). 如果改用 Kim TOH 写的 mexeig 做 [V,D]=mexeig(A), 计算时间大为节省.

特征值分解使用 mexeig 的迭代次数和计算时间

| $n \times n$ Matrix | Classical PPA | | Extended PPA | |
|---|---|---|---|---|
| $n =$ | No. It | CPU Sec. | No. It | CPU Sec. |
| 100 | 30 | 0.12 | 23 | 0.10 |
| 200 | 33 | 0.54 | 25 | 0.40 |
| 500 | 38 | 7.99 | 26 | 6.25 |
| 800 | 38 | 37.44 | 28 | 27.04 |
| 1000 | 45 | 94.32 | 30 | 55.32 |
| 2000 | 62 | 723.40 | 38 | 482.18 |



## 4.2 矩阵完整化方面的应用

设 $M$ 是一个 $m \times n$ 矩阵, $\Pi$ 是矩阵元素的指标集.

$$\Pi = \{(ij) \,|\, i \in \{1, 2, \ldots, m\}, \; j \in \{1, 2, \ldots, n\}\}.$$

矩阵完整化问题是 **由部分信息获取全部信息**

- 显然, 没有其他信息, 恢复一个一般的矩阵是不可能的.
- 幸运的是, 在许多情况下, 我们要恢复的矩阵是一个低秩矩阵.

恢复低秩矩阵的一般数学模型是:

$$\boxed{\min\{\textbf{rank}(X) \;:\; X_{ij} = M_{ij}, \;\; (ij) \in \Pi\}.}$$

然而, 这个问题是NP-hard 的. 根据 **Candés, Recht, Tao** 最近的工作

- E. J. Candés and B. Recht, Exact Matrix Completion via Convex Optimization, 2008.
- E. J. Candés and T. Tao, The Power of Convex Relaxation: Near-Optimial Matrix Completion, 2009,

在适当(实际问题具备的)条件下, 大多数不完整信息的低秩矩阵可以通过求解松弛问题

$$\min\{\|X\|_* \mid X_{ij} = M_{ij},\ (ij) \in \Pi\} \tag{4.5}$$

得到精确恢复. 其中 $\|X\|_*$ 表示矩阵 $X$ 的奇异值的和. 通常称为矩阵 $X$ 的核模— Nuclear Norm. 这类问题的商业应用, 可见 [10].

问题 (4.5) 是形如 (2.1) 的等式约束凸优化问题, 其中 $\|A^TA\| = 1$. 我们将 (4.5) 的等式约束记为 $X_\Pi = M_\Pi$，并用 $Z \in \Re^{m\times n}$ 作为相应的 Lagrange 乘子.

**Dual-Primal PPA 求解问题(4.5)**

对给定的 $(X^k, Z^k)$, 用 (3.10)–(3.11) 产生$(\tilde{X}^k, \tilde{Z}^k)$:

1. Producing $\tilde{Z}^k$ by

$$\tilde{Z}^k_\Pi = Z^k_\Pi - \frac{1}{s}(X^k_\Pi - M_\Pi).$$

2. Finding $\tilde{X}^k$ which is the solution of the following linear variational inequality

$$\min\ \{\|X\|_* + \frac{r}{2}\|X - [X^k + \frac{1}{r}(2\tilde{Z}^k_\Pi - Z^k_\Pi)]\|_F^2\} \tag{4.6}$$



**子问题 (4.6) 求解的具体做法:** 只需考虑如何求解

$$\tilde{X}^k = \text{Argmin}\{\frac{1}{r}\|X\|_* + \frac{1}{2}\|X - A\|_F^2\}. \tag{4.7}$$

我们将 $A$ 做 SVD 分解

$$A = U\Lambda V^T,$$

并代入 (4.7), 得到

$$\frac{1}{r}\|\tilde{X}\|_* + \frac{1}{2}\|\tilde{X} - U\Lambda V^T\|_F^2 = \frac{1}{r}\|U^T\tilde{X}V\|_* + \frac{1}{2}\|U^T\tilde{X}V - \Lambda\|_F^2.$$

上面的等式是由于矩阵的奇异值及$F$-范数在正交变换下不变的原因. 因此, $U^T\tilde{X}V$ 应该是一个非负的对角矩阵. 设 $U^T\tilde{X}V = \tilde{\Lambda}$, 换句话说,

$$\tilde{X} = U\tilde{\Lambda}V^T. \tag{4.8}$$

对给定的非负对角矩阵 $\Lambda$,

$$\min\{\frac{1}{r}\|\tilde{\Lambda}\|_* + \frac{1}{2}\|\tilde{\Lambda} - \Lambda\|_F^2\}$$

的解对角矩阵 $\tilde{\Lambda}$, 其对角元通过

$$\tilde{\lambda}_j = \lambda_j - \min(\lambda_j, \frac{1}{r}), \tag{4.9}$$

就能得到. 代入 (4.8) 就得到 (4.7) 的解 $\tilde{X}^k$. 因此, 每次迭代的主要工作量是做一个矩阵的 SVD 分解.

如果以 $\lambda$ 和 $\tilde{\lambda}$ 分别表示对角矩阵 $\Lambda$ 和 $\tilde{\Lambda}$ 的对角元生成的向量, 由于 $\lambda$ 是非负向量, 关系式 (4.9) 也可以写成

$$\tilde{\lambda} = \lambda - P_{B_\infty^{1/r}}[\lambda],$$

其中 $B_\infty^{1/r}$ 是无穷模下半径为 $1/r$ 的“圆”（一个立方体）. 上述运算在文献中常常称作 **Shrinkage**.

**结论**: 将线性约束的凸优化问题转换成单调变分不等式, 再用本文介绍的 PPA 方法求解, 每次迭代中要求解的子问题, 数值代数中都有确定的成熟的方法求解 !



**数值试验** 数值试验例子取自[1]

一个秩为 $ra$ 的 $n \times n$ 的自由度是 $d_{ra} := ra(2n - ra)$.

**生成试验问题:**

- 先用高斯同分布 (Gaussian i.i.d) 独立生成两个 $n \times ra$ 的矩阵 $M_1$ 和 $M_2$, 然后令$M = M_1 M_2^T$, 则 $n \times n$ 矩阵 $M$ 的秩为 $ra$.
- 随机选定 $M$ 的 $m$ 个元素作为已知元素, 这些元素的下标集为 $\Pi$.

**计算结果:**

- 矩阵完整化问题的难度与比率 $m/d_{ra}$ 和 $m/n^2$ 都有关系.
- 分别用 Classical PPA 和 Extended PPA ($\gamma = 1.5$) 进行计算.
- 正定矩阵 $G$ (see (3.9)) 中的参数 $r$, $s$ 分别取 $rs = 1.01$ 和 $r = 0.005$.
- 停机准则采用相对误差 $\|X_\Pi^k - M_\Pi\|_F / \|M_\Pi\|_F \le 10^{-4}$.

我们用 $\gamma = 1.5$ 的 Extended PPA 求解, 注意到 KKT 条件 Primal 部分((3.8) 的上半部分) 的不满足量是

$$r(\tilde{X}^k - X^k) - (\tilde{Z}^k - Z^k).$$

♣ 矩阵完整化的程序在附件的 Codes-04 的文件夹“*矩阵完整化*”中. 只要运行 demo.m 就可以了. 要对不同情形试验, 只要在 demo.m 中用 % 做适当选择. 其中的 PPAC.m 和 PPAE.m 分别是 Classical PPA 和 Extended PPA 的子程序.

## Code 4.b. Creating the test examples of Matrix Completion

```
%%  Creating the test examples of the matrix Completion problem     %(1)
clear all;  clc                                                     %(2)
maxIt=100;         tol = 1e-4;                                      %(3)
r=0.005;           s=1.01/r;        gamma=1.5;                      %(4)
  n=200;             ra = 10;         oversampling = 5;             %(5)
% n=1000;   ra=100;     oversampling = 3; %%  Iteration No. 31      %(6)
% n=1000;   ra=50;      oversampling = 4; %%  Iteration No. 36      %(7)
% n=1000;   ra=10;      oversampling = 6; %%  Iteration No. 78      %(8)
%% Generating the test problem                                      %(9)
rs = randseed;        randn('state',rs);                           %(10)
M=randn(n,ra)*randn(ra,n);          %% The matrix will be completed %(11)
df =ra*(n*2-ra);                    %% The freedom of the matrix    %(12)
mo=oversampling;                                                   %(13)
m =min(mo*df,round(.99*n*n));        %% No. of the known elements  %(14)
Omega= randsample(n^2,m);            %% Define the subset Omega    %(15)
fprintf('Matrix: n=%4d  Rank(M)=%3d  Oversampling=%2d \n',n,ra,mo);%(16)
```



## Code 4.3. Extended PPA for Matrix Completion Problem

```
function PPAE(n,r,s,M,Omega,maxIt,tol,gamma)     %  Ititial Process %%(1)
X=zeros(n);     Y=zeros(n);      YT=zeros(n);                        %(2)
  nM0=norm(M(Omega),'fro');     eps=1;  VioKKT=1;   k=0;   tic;      %(3)
%% Minimum nuclear norm solution by PPA method                       %(4)
while  (eps > tol && k<= maxIt)                                      %(5)
 if mod(k,5)==0                                                      %(6)
 fprintf('It=%3d |X-M|/|M|=%9.2e VioKKT=%9.2e\n',k,eps,VioKKT); end;%(7)
  k=k+1;        X0=X;       Y0=Y;                                    %(8)
  YT(Omega)=Y0(Omega)-(X0(Omega)-M(Omega))/s;             EY=Y-YT;   %(9)
  A = X0 + (YT*2-Y0)/r;                   [U,D,V]=svd(A,0);         %(10)
  D=D-eye(n)/r;      D=max(D,0);      XT=(U*D)*V';        EX=X-XT;  %(11)
  DXM=XT(Omega)-M(Omega);             eps = norm(DXM,'fro')/nM0;    %(12)
  VioKKT = max( max(max(abs(EX)))*r, max(max(abs(EY))) );           %(13)
   if (eps <= tol)   gamma=1;   end;                                %(14)
  X = X0 - EX*gamma;                                                %(15)
  Y(Omega) = Y0(Omega) - EY(Omega)*gamma;                           %(16)
end;                                                                %(17)
 fprintf('It=%3d  |X-M|/|M|=%9.2e  Vi0KKT=%9.2e \n',k,eps,VioKKT); %(18)
   RelEr=norm((X-M),'fro')/norm(M,'fro');    toc;                   %(19)
 fprintf(' Relative error = %9.2e  Rank(X)=%3d \n',RelEr,rank(X)); %(20)
 fprintf(' Violation of KKT Condition = %9.2e  \n',VioKKT);         %(21)
```

矩阵完整化问题：用 Matlab 中标准 SVD 求解结果

| Unknown $n \times n$ matrix $M$ | | | | Computational Results | | | |
|---|---|---|---|---|---|---|---|
| $n$ | rank$(ra)$ | $m/d_{ra}$ | $m/n^2$ | #iters | times(Sec.) | relative error | KKT-Violation |
| $1000$ | 10 | 6 | 0.12 | 76 | 841.59 | 9.38E-5 | 9.31E-6 |
| $1000$ | 50 | 4 | 0.39 | 37 | 406.24 | 1.21E-4 | 2.11E-5 |
| $1000$ | 100 | 3 | 0.58 | 31 | 362.58 | 1.50E-4 | 2.88E-5 |

♣ 用 Matlab 中的 SVD, 做一次 SVD 的花费很大, 总耗时与迭代次数成比例.

♣ 用 PROPACK [8] 中的 SVD, 快许多, 总耗时主要与问题性质有关. 我们主要对迭代次数感兴趣, 只报道用 Matlab 做 SVD 的结果, 也附上所需要的子程序.

对矩阵完整化问题, (3.4) 和 (3.8) 的下半部分误差用$\|X^k_\Pi - M_\Pi\|_F/\|M_\Pi\|_F$ 控制. 由于 (3.4) 和 (3.8) 的上半部分是

$$\theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T\{(-A^T\tilde{\lambda}^k) + [r(\tilde{x}^k - x^k) \pm A^T(\tilde{\lambda}^k - \lambda^k)]\},$$

因此上半部分误差项是 $[r(\tilde{x}^k - x^k) \mp A^T(\tilde{\lambda}^k - \lambda^k)]$. 对矩阵完整化问题, $A$是投影矩阵, 其模为 $1$, 所以我们在计算结果中也列出

$$\text{KKT-Violation} := \max\{r\max_{ij}|X^k_{ij} - \tilde{X}^k_{ij}|,\ \max_{ij}|Z^k_{ij} - \tilde{Z}^k_{ij}|\}.$$

♣ 论文 [1] 是最早发表在 SIAM J. Optimization 上的求解矩阵完整化问题的文章. 对以上试验的三个例子, [1] 中的方法达到同样精度要求的迭代次数分别是 117, 114 和 129 (See the first three examples in Table 5.1 of [1], pp. 1974), 每次迭代的主要工作量也是做一次 SVD 分解. 由于采用了不完全分解技术, [1] 中节省了 SVD 分解的时间. 我们调用的是 Matlab 中标准SVD, 虽然花费了更少的迭代次数, 但没有节省总的运行时间.



## 4.3 压缩传感(Compressed Sensing)问题

- 压缩传感问题是利用信号的稀疏性质, 只对其进行相对较少的测量, 然后能以较小的误差或者精确地恢复原始信号。
- CS 问题: 求满足 $Ax = b$ 的非零元个数最小的解

$$\min\{\|x\|_0 \mid Ax = b\}.$$

其中$\|x\|_0$ 表示向量 $x$ 的非零元的个数, 它不是常规意义下的模。是一个很难求解的组合优化问题.

$$\boxed{\|x\|_1 \le \|x\|_0,\ \forall\, x \in \Omega = \{x \in R^n \mid |x_j| \le 1,\ j = 1, \ldots, n\}.}$$

Candés 和 Tao 证明了在一定条件下, 问题

$$\min\{\|x\|_0\ :\ Ax = b\} \qquad \text{与问题} \qquad \min\{\|x\|_1\ :\ Ax = b\}$$

的解在概率意义下是相同的.

极小化问题

$$\min_x \{\, \mu\|x\|_1 \mid Ax = b \,\} \qquad \text{(BP)} \qquad (4.10)$$

是形如 (2.1) 的等式约束凸优化问题.

**PPA 算法求解问题(4.10)**

对给定的 $(x^k, \lambda^k)$, 用 (3.10)–(3.11) 产生$(\tilde{x}^k, \tilde{\lambda}^k)$:

1. Producing $\tilde{\lambda}^k$ by
$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}(Ax^k - b).$$
2. Finding $\tilde{x}^k$ which is the solution of the following minimization problem
$$\min\{\, \mu\|x\|_1 + \frac{r}{2}\|x - [x^k + \frac{1}{r}A^T(2\tilde{\lambda}^k - \lambda^k)]\|^2\}. \qquad (4.11)$$

**子问题 (4.11) 求解的具体做法:** 我们只要考虑如何求解

$$\min_{x\in\Re^n} \{\tau\|x\|_1 + \frac{1}{2}\|x - a\|^2\}. \qquad (4.12)$$



设 $x^* \in \Re^n$ 是 (4.12) 的解,

- 如果 $x_j^* > 0$, 对 $\tau x + \frac{1}{2}\|x - a\|^2$ 求导, 有
$$\tau + x_j^* - a_j = 0, \quad 即 \quad x_j^* = a_j - \tau, \qquad (这隐含了 \quad a_j > \tau).$$
- 如果 $x_j^* < 0$, 对 $-\tau x + \frac{1}{2}\|x - a\|^2$ 求导, 有
$$-\tau + x_j^* - a_j = 0, \quad 即 \quad x_j^* = a_j + \tau, \qquad (这隐含了 \quad a_j < -\tau).$$
- 为什么 $x_j^* = 0$, 是因为 $-\tau \le a_j \le \tau$.

综上所述, 问题 (4.12) 的解由

$$x_j^* = \begin{cases} a_j - \tau, & \text{if} \quad a_j > \tau \\ a_j + \tau, & \text{if} \quad a_j < -\tau \\ 0, & \text{if} \quad -\tau \le a_j \le \tau. \end{cases}$$

给出. 这也可以写成 Shrinkage 的形式

$$x^* = a - P_{B_\infty^\tau}[a], \qquad \text{where} \qquad B_\infty^\tau = \{\xi \in \Re^n \mid -\tau e \le \xi \le \tau e\}.$$

实际 CS 问题的计算, 要加进一些 Continuation 技术, 限于篇幅, 不详细介绍.

## 4.4 Min-Max 问题上的应用

第一讲 §4.2 中提到用全变差极小处理图像去模糊 [2], 经离散化以后, 问题的数学模型是的 min-max 问题,

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \ \Phi(x,y) := \theta_1(x) + y^TAx - \theta_2(y). \tag{4.13}$$

它可以转换成等价的变分不等式: 求 $(x^*,y^*)\in\mathcal{X}\times\mathcal{Y}$, 使得

$$\begin{pmatrix} x-x^* \\ y-y^* \end{pmatrix}^T \begin{pmatrix} f(x^*)+A^Ty^* \\ g(y^*)-Ax^* \end{pmatrix} \ge 0, \quad \forall (x,y)\in\mathcal{X}\times\mathcal{Y}, \tag{4.14}$$

其中 $f(x)\in\partial\theta_1(x)$, $g(y)\in\partial\theta_2(y)$. 文献 [6] 中提到用这里的 PPA 方法去求解. 对给定的 $(x^k,y^k)$, 求 $(\tilde{x}^k,\tilde{y}^k)\in\mathcal{X}\times\mathcal{Y}$, 对一切 $(x,y)\in\mathcal{X}\times\mathcal{Y}$, 都有

$$\begin{pmatrix} x-\tilde{x}^k \\ y-\tilde{y}^k \end{pmatrix}^T \left\{ \begin{pmatrix} f(\tilde{x}^k)+A^T\tilde{y}^k \\ g(\tilde{y}^k)-A\tilde{x}^k \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k-x^k)+A^T(\tilde{y}^k-y^k) \\ +A(\tilde{x}^k-x^k)+s(\tilde{y}^k-y^k) \end{pmatrix} \right\} \ge 0. \tag{4.15}$$



注意到 (4.15) 的下半部分中只含有未知的 $\tilde{y}^k$. 它可以通过求解 $y$ 的子问题:

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + y^TAx^k + \frac{s}{2}\|y-y^k\|^2 \,|\, y\in\mathcal{Y}\}. \tag{4.16a}$$

得到. 然后, 为求得 (4.15) 上半部分得 $\tilde{x}^k$, 我们只要求解一个关于 $x$ 的问题:

$$\tilde{x}^k = \text{Argmin}\Big\{\theta_1(x) + \frac{r}{2}\Big\|x - \big[x^k - \frac{1}{r}A^T(2\tilde{y}^k - y^k)\big]\Big\|^2 \,\Big|\, x\in\mathcal{X}\Big\}. \tag{4.16b}$$

当然, PPA 预测点 $(\tilde{x}^k,\tilde{y}^k)\in\mathcal{X}\times\mathcal{Y}$ 也可以通过先 $x$, 后 $y$ 的顺序完成.

对给定的 $(x^k,y^k)$, 通过求解 $x$ 的子问题

$$\tilde{x}^k = \text{Argmin}\{\theta_1(x) + x^TA^Ty^k + \frac{r}{2}\|x-x^k\|^2 \,|\, x\in\mathcal{X}\}. \tag{4.17a}$$

得到 $\tilde{x}^k$, 然后求 $\tilde{y}^k$ 通过一个关于 $y$ 的子问题:

$$\tilde{y}^k = \text{Argmin}\Big\{\theta_2(y) + \frac{s}{2}\Big\|y - \big[y^k + \frac{1}{s}A(2\tilde{x}^k - x^k)\big]\Big\|^2 \,\Big|\, y\in\mathcal{Y}\Big\}. \tag{4.17b}$$

由 (4.17) 生成的 $(\tilde{x}^k, \tilde{y}^k) \in \mathcal{X} \times \mathcal{Y}$, 对一切 求 $(x, y) \in \mathcal{X} \times \mathcal{Y}$, 都有

$$\begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \end{pmatrix}^T \left\{ \begin{pmatrix} f(\tilde{x}^k) + A^T \tilde{y}^k \\ g(\tilde{y}^k) - A\tilde{x}^k \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) - A^T(\tilde{y}^k - y^k) \\ -A(\tilde{x}^k - x^k) + s(\tilde{y}^k - y^k) \end{pmatrix} \right\} \ge 0. \tag{4.18}$$

无论是 (4.15) 还是 (4.18), 都可以写成

$$\tilde{u}^k \in \Omega, \qquad (u - \tilde{u}^k)^T \{F(\tilde{u}^k) + G(\tilde{u}^k - u^k)\} \ge 0, \quad \forall\, u \in \Omega, \tag{4.19}$$

的形式, 所不同的只是

$$G = \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix} \qquad \text{和} \qquad G = \begin{pmatrix} rI_n & -A^T \\ -A & sI_m \end{pmatrix}.$$

它们都是当 $rs > \|A^T A\|$ 时候正定. 我们建议新的迭代点$u^{k+1}$ 用迭代式

$$u^{k+1} = u^k - \gamma(u^k - \tilde{u}^k), \quad \gamma \in [1, 2),$$

生成, 一般取 $\gamma = 1.5$. 对一些工程技术上的问题, 采取 $\gamma > 1$ 的迭代满足停机准则以后, 最后加一次 $\gamma = 1$ 的迭代. 这样既提高了速度, 又满足实际问题对诸如 Low-Rank 这样的要求. 这类方法在图像处理中的应用可见论文 [6, 7].

定制的 PPA 算法具有 $O(1/k)$ 的收敛速率, 证明可以在论文 [4] 中找到.



# References


[1] J. F. Cai, E. J. Candès and Z. W. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. **20**, 1956-1982, 2010.

[2] A. Chambolle and T. Pock, A first-order primal-dual algorithms for convex problem with applications to imaging, J. Math. Imaging Vison, **40**, 120-145, 2011.

[3] B. S. He, X. L. Fu, Z.K. Jiang, Proximal point algorithm using a linear proximal term, JOTA **141**, 209-239, 2009.

[4] B. S. He and Y. Shen, On the convergence rate of customized proximal point algorithm for convex optimization and saddle-point problem (in Chinese). Sic Sin Math, **42**, 515-525, 2012. doi: 10.1360/012011-1049

[5] B. S. He and M.-H. Xu, A general framework of contraction methods for monotone variational inequalities, Pacific J. Optimization, **4**, 195-212, 2008.

[6] B.S. He and X. M. Yuan, Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective, SIAM Journal on Imaging Science, **5**, 119-149, 2012.

[7] B. S. He, X. M. Yuan and W.X. Zhang, A customized proximal point algorithm for convex minimization with linear constraints, Comput. Optim. Appl., 56: 559-572, 2013.

[8] R. M. Larsen, Propack – software for large and sparse svd calculation, http://sun.stanford.edu/rmunk/PROPACK/, 2005.

[9] B. Martinet, Regularisation, d'inéquations variationelles par approximations succesives, Rev. Francaise d'Inform. Recherche Oper., **4**, 154-159, 1970.

[10] Netflix, Netflix Prize, Available at http://www.netflixprize.com/. 2006.

[11] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Cont. Optim. **14**, 877-898, 1976.

[12] M. Zhu and T. F. Chan, An efficient primal-dual hybrid gradient algorithm for total variation image restoration, CAM Reports 08-34, UCLA, 2008.

# 凸优化和单调变分不等式的收缩算法

## 第五讲: 线性约束凸优化问题
## 基于松弛 PPA 的收缩算法

Relaxed PPA-based Contraction Methods for
Linearly Constrained Convex Optimization

南京大学数学系　　何炳生
hebma@nju.edu.cn

The context of this lecture is based on the publications [6]



## 线性约束的凸优化问题

这一讲要求解的问题是

$$\boxed{\min\{\theta(x) \mid Ax = b\,(or\ Ax \ge b)\ x \in \mathcal{X}\}}$$

其中 $\theta(x)$ 是凸函数, $A \in \Re^{m\times n}$, $b \in \Re^m$, $\mathcal{X}$ 是 $\Re^n$ 中的闭凸集

对任意给定的 $r > 0$ 和 $a \in \Re^n$, 通篇我们假设子问题

$$\boxed{\min\,\{\theta(x) + \tfrac{r}{2}\|x - a\|^2 \,|\, x \in \mathcal{X}\}}$$

的求解是简单的.

**目的** 将线性约束的凸优化问题转换成单调变分不等式, 再用自调比方法, 生成类似于 PPA 的正则项. 我们并不要求正则项系数矩阵 $Q$ 对称正定, 只要在迭代中都有 $(u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k) \ge \tau\|u^k - \tilde{u}^k\|^2$. 基于松弛 PPA 的收缩算法是一类预测一校正方法, 产生预测点都用松弛 PPA 方法.

# 1 线性约束凸优化问题的 PPA 算法

## 1.1 线性约束凸优化问题与单调变分不等式

线性约束的凸优化问题

$$\min\{\theta(x) \mid Ax = b\,(or\ Ax \ge b)\ x \in \mathcal{X}\} \tag{1.1}$$

的 Lagrange 函数是定义在 $\mathcal{X} \times \Lambda$ 上的

$$L(x, \lambda) = \theta(x) - \lambda^T (Ax - b),$$

其中

$$\Lambda = \begin{cases} \Re^m, & \text{for the equality constraints} \quad Ax = b, \\ \Re^m_+, & \text{for the inequality constraints} \quad Ax \ge b. \end{cases}$$

设 $(x^*, \lambda^*)$ 是 Lagrange 函数的一个鞍点, 便有

$$L_{\lambda \in \Lambda}(x^*, \lambda) \le L(x^*, \lambda^*) \le L_{x \in \mathcal{X}}(x, \lambda^*).$$



求 Lagrange 函数的一个鞍点等价于求 $(x^*, \lambda^*)$ 使其满足:

$$\begin{cases} x^* \in \mathcal{X}, & \theta(x) - \theta(x^*) + (x - x^*)^T (-A^T \lambda^*) \ge 0, \quad \forall\, x \in \mathcal{X}, \\ \lambda^* \in \Lambda, & (\lambda - \lambda^*)^T (Ax^* - b) \ge 0, \quad \forall\, \lambda \in \Lambda. \end{cases} \tag{1.2}$$

通过定义

$$u = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F(u) = \begin{pmatrix} -A^T \lambda \\ Ax - b \end{pmatrix} \tag{1.3}$$

和

$$\Omega = \mathcal{X} \times \Lambda, \tag{1.4}$$

求解 (1.1) 相当于求解混合变分不等式

$$\text{VI}(\Omega, F, \theta) \quad u^* \in \Omega, \quad \theta(x) - \theta(x^*) + (u - u^*)^T F(u^*) \ge 0, \quad \forall\, u \in \Omega. \tag{1.5}$$

注意到 $\theta(x)$ 是凸函数, 此外 (1.3) 中的算子 $F$ 是仿射单调的.

**我们通过求解相应的混合单调变分不等式来求解线性约束的凸优化问题.**

## 1.2 $G$-模下的 PPA 算法和松弛 PPA 的收缩算法

邻近点算法 (PPA) 是求解混合单调变分不等式的一个经典方法. 用邻近点算法求解问题 (1.5), 对给定的 $u^k \in \Omega$ 和 $r > 0$, $k$-次迭代是求 $\tilde{u}^k \in \Omega$, 使得

$$\theta(x) - \theta(\tilde{x}^k) + (u - \tilde{u}^k)^T \{F(\tilde{u}^k) + r(\tilde{u}^k - u^k)\} \ge 0, \quad \forall u \in \Omega. \qquad (1.6)$$

我们取 $\gamma \in [1, 2)$, 用公式 $u^{k+1} = u^k - \gamma(u^k - \tilde{u}^k)$ 产生下一个迭代点. 序列 $\{u^k\}$ 满足关系式

$$\|u^{k+1} - u^*\|^2 \le \|u^k - u^*\|^2 - \gamma(2-\gamma)\|u^k - \tilde{u}^k\|^2.$$

对解集 $\Omega^*$, 经典的 PPA 算法产生的序列 $\{u^k\}$ 是 Fejér 单调的.

♣ 直接求解 PPA 子问题 (1.6) 一般来说是办不到的. 为求解与凸优化问题 (1.1) 等价的变分不等式 (1.5), 上一讲提出的 Customized PPA 方法的子问题是

$$\tilde{u}^k \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + (u - \tilde{u}^k)^T \{F(\tilde{u}^k) + G(\tilde{u}^k - u^k)\} \ge 0, \quad \forall u \in \Omega, \qquad (1.7)$$

其中

$$G = \begin{pmatrix} rI_n & -A^T \\ -A & sI_m \end{pmatrix}.$$



注意到与讨论的优化问题等价的变分不等式 (1.2) 的具体结构 (见 (1.3) 式), 子问题 (1.7) 便是求 $(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega$, 使得对一切 $(x, \lambda) \in \Omega$, 都有

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) - A^T(\tilde{\lambda}^k - \lambda^k) \\ -A(\tilde{x}^k - x^k) + s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \ge 0, \quad \forall (x, \lambda) \in \Omega. \qquad (1.8)$$

上式下半部分中未知的 $\tilde{x}^k$ 可以消掉, 问题 (1.8) 的一半松弛成

$$\tilde{\lambda}^k \in \Lambda, \quad (\lambda - \tilde{\lambda}^k)^T \{(Ax^k - b) + s(\tilde{\lambda}^k - \lambda^k)\} \ge 0, \quad \forall \lambda \in \Lambda.$$

这个 $\tilde{\lambda}^k$ 可以通过 $\tilde{\lambda}^k = P_\Lambda[\lambda^k - \frac{1}{s}(Ax^k - b)]$ 直接给出. 有了 $\tilde{\lambda}^k$, 通过求解极小化问题

$$\min \left\{ \theta(x) + \frac{r}{2} \left\| x - \left[x^k + \frac{1}{r} A^T(2\tilde{\lambda}^k - \lambda^k)\right] \right\|^2 \,\middle|\, x \in \mathcal{X} \right\} \qquad (1.9)$$

得到的 $\tilde{x}^k$ 满足 (1.8) 的上半部分. 因为 (1.9) 中的 $\tilde{\lambda}^k$ 也是已知的, 极小化问题 (1.9) 的类型根据假设是容易求解的. 用公式 $u^{k+1} = u^k - \gamma(u^k - \tilde{u}^k)$ 产生

的迭代序列 $\{u^k\}$ 满足

$$\|u^{k+1} - u^*\|_G^2 \le \|u^k - u^*\|_G^2 - \gamma(2-\gamma)\|u^k - \tilde{u}^k\|_G^2.$$

说到底, 这个 $G$-模下的 PPA 算法, 也是对经典的 PPA 松弛后得到的收缩算法.

上述 $G$-模下 PPA 算法子问题 (1.8) 中的参数 $r, s$ 要满足 $rs > \|A^TA\|$, 矩阵 $G$ 才是正定的, 这才保证算法收敛. 有时估计 $\|A^TA\|$ 并不容易, 过大的估计相当于正则项太大, 会影响总体收敛速度. 这一讲试图解决这些问题.

## 1.3 松弛条件下 PPA 收缩算法

**Relaxed PPA 的想法** 为求解变分不等式 (1.2), 在第 $k$-次迭代中, 经典的 PPA 是要求 $(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega$ 使得

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \ge 0, \ (x, \lambda) \in \Omega. \tag{1.10}$$



因为上面的子变分不等式的上下两部分都含有未知的 $\tilde{x}^k$ 和 $\tilde{\lambda}^k$, 直接求解一般办不到. 如果将上半部分中的 $\tilde{\lambda}^k$ 松弛成 $\lambda^k$, 则变成求 $(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega$ 使得

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\lambda^k \\ A\tilde{x}^k - b \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \ge 0, \ (x, \lambda) \in \Omega. \tag{1.11}$$

我们把子问题 (1.11) 称为松弛的 PPA (Relaxed PPA). 松弛以后的问题 (1.11) 是容易求解的. 首先 (1.11) 的上半 (Primal) 部分可以通过求解极小化问题

$$\min\Big\{\theta(x) + \frac{r}{2}\big\|x - \big[x^k + \frac{1}{r}A^T\lambda^k\big]\big\|^2 \,\big|\, x \in \mathcal{X}\Big\}$$

得到. 根据假设条件, 这是一个容易求解的问题. 有了 $\tilde{x}^k$, (1.11) 的下半 (Dual) 部分中要求的只剩下 $\tilde{\lambda}^k$, 它的数学形式是

$$\tilde{\lambda}^k \in \Lambda, \quad (\lambda - \tilde{\lambda}^k)^T\{(A\tilde{x}^k - b) + s(\tilde{\lambda}^k - \lambda^k)\} \ge 0, \quad \forall \lambda \in \Lambda.$$

根据变分不等式和投影的关系, $\tilde{\lambda}^k$ 可以通过

$$\tilde{\lambda}^k = P_\Lambda[\lambda^k - \frac{1}{s}(A\tilde{x}^k - b)]$$

直接给出. 我们把 (1.11) 写成:

$$(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} \right.$$

$$\left. + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ 0 \end{pmatrix} + \begin{pmatrix} A^T(\tilde{\lambda}^k - \lambda^k) \\ s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \geq 0, \quad \forall (x, \lambda) \in \Omega.$$

它的紧凑形式是

$$\tilde{u}^k \in \Omega, \ \theta(x) - \theta(\tilde{x}^k) + (u - \tilde{u}^k)^T \{F(\tilde{u}^k) + Q(\tilde{u}^k - u^k)\} \geq 0, \ \forall u \in \Omega, \quad (1.12)$$

其中矩阵

$$Q = \begin{pmatrix} rI_n & A^T \\ 0 & sI_m \end{pmatrix}$$

是非对称的. 换句话说, 这样的检验点 (或预测点) $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k)$ 是通过对经典的 PPA 子问题松弛求得的. 由于 (1.12) 中除了 $F(\tilde{u}^k)$ 外就是 $Q(\tilde{u}^k - u^k)$, 也可以把由此建立的收缩算法称为基于线性临近点项的 PPA 收缩算法. 收敛证明的参考文献可见:



B. S. He, X. L. Fu and Z.K. Jiang, Proximal point algorithm using a linear proximal term, JOTA 141：299-319, 2009.

上述论文告诉我们, 基于线性临近点项的 PPA, 只要 (1.12) 中的矩阵 $Q$ 满足

$$Q^T + Q = \begin{pmatrix} 2rI_n & A^T \\ A & 2sI_m \end{pmatrix} \succ 0,$$

就可以构造收缩算法. 由上式可知, 当 $rs > \frac{1}{4}\|A^TA\|$ 时, $Q^T + Q$ 就正定. 这个条件显然比前一讲 $G$-模下的 PPA 算法对参数 $r, s$ 要求 $rs > \|A^TA\|$ 来得宽松. 这是我们这一讲继续 Relaxed PPA 研究的理由.

**在统一框架下的考虑:** 对给定的 $u^k$, 我们生成 $\tilde{u}^k \in \Omega$ 使得(1.12) 成立, 其中 $Q$ 是一个矩阵. 并不要求 $Q$ 对称, 只要求存在一个 $\tau > 0$, 有

$$(u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k) \geq \tau \|u^k - \tilde{u}^k\|^2, \ \ \forall\, k \geq 0.$$

上述条件在 $Q^T + Q$ 正定时一定成立.

# 2 Primal-Dual 松弛 PPA 收缩算法

用松弛 PPA 按先 $\tilde{x}^k$ (primal) 后 $\tilde{\lambda}^k$ (dual) 的顺序生成预测点 $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k)$, 然后构造的收缩算法, 称为基于Primal-Dual 松弛 PPA 的收缩算法 (Primal-dual relaxed PPA based contraction method).

## 2.1 Primal-Dual 生成预测点

**Primal–Dual Method 生成预测点** 对给定的 $(x^k, \lambda^k)$ 和 $r > 0$, 通过求解

$$\min \left\{\theta(x) + \frac{r}{2}\left\|x - \left[x^k + \frac{1}{r}A^T\lambda^k\right]\right\|^2 \,\middle|\, x \in \mathcal{X}\right\} \tag{2.1a}$$

得到 Primal 预测点 $\tilde{x}^k$. 再选取适当的 $s > 0$ 并用

$$\tilde{\lambda}^k = P_\Lambda[\lambda^k - \frac{1}{s}(A\tilde{x}^k - b)] \tag{2.1b}$$

生成 Dual 预测点 $\tilde{\lambda}^k$ (如何选取 $s > 0$ 放在后面讨论).



注意到用 (2.1a) 生成的 Primal 预测点 $\tilde{x}^k$ 满足

$$\tilde{x}^k \in \mathcal{X}, \ \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T\{r(\tilde{x}^k - x^k) - A^T\lambda^k\} \ge 0, \ \forall x \in \mathcal{X}. \tag{2.2}$$

将 (2.1b) 改写成 $\tilde{\lambda}^k = P_\Lambda\{\tilde{\lambda}^k - [(\tilde{\lambda}^k - \lambda^k) + \frac{1}{s}(A\tilde{x}^k - b)]\}$, 则有

$$\tilde{\lambda}^k \in \Lambda, \ \ (\lambda - \tilde{\lambda}^k)^T\{(A\tilde{x}^k - b) + s(\tilde{\lambda}^k - \lambda^k)\} \ge 0, \ \ \forall \lambda \in \Lambda. \tag{2.3}$$

把 (2.2) 和 (2.3) 写在一起, 有

$$\begin{aligned}(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \Bigg\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} \\ + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ 0 \end{pmatrix} + \begin{pmatrix} A^T(\tilde{\lambda}^k - \lambda^k) \\ s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \Bigg\} \ge 0, \quad \forall (x, \lambda) \in \Omega.\end{aligned}$$

它的紧凑形式是

$$\tilde{u}^k \in \Omega, \ \theta(x) - \theta(\tilde{x}^k) + (u - \tilde{u}^k)^T\{F(\tilde{u}^k) + Q(\tilde{u}^k - u^k)\} \ge 0, \ \forall u \in \Omega, \tag{2.4}$$

其中矩阵 $Q$ (对应于 Primal-Dual 方法生成预测点的 $Q$, 有时也记成 $Q_{PD}$)

$$Q_{PD} = \begin{pmatrix} rI_n & A^T \\ 0 & sI_m \end{pmatrix}, \tag{2.5}$$

是非对称的.

如果 (2.4) 中的矩阵 $Q$ 是对称正定的, 这个生成预测点 $\tilde{u}^k$ 的方法就是 PPA 算法. 由于这里的 $Q$ 是非对称矩阵, 按照 [6] 中的说法, 这里的 Relaxed PPA 是带线性临近点项的 PPA.

以 $u^*$ 替代 (2.4) 中的 $u \in \Omega$, 可得到

$$(\tilde{u}^k - u^*)^T Q(u^k - \tilde{u}^k) \geq \theta(\tilde{x}^k) - \theta(x^*) + (\tilde{u}^k - u^*)^T F(\tilde{u}^k). \tag{2.6}$$

因为 $\tilde{u}^k \in \Omega$ 并且 $u^*$ 是 VI$(\Omega, F)$ 的解, 根据混合变分不等式 (1.5) 的定义, 有

$$\theta(\tilde{x}^k) - \theta(x^*) + (\tilde{u}^k - u^*)^T F(u^*) \geq 0.$$

由于 $F$ 是单调算子, 因此有

$$(\tilde{u}^k - u^*)^T F(\tilde{u}^k) \geq (\tilde{u}^k - u^*)^T F(u^*).$$



根据上面二个关系式得到 (2.6) 的右端非负. 我们定义

$$\varphi(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k). \tag{2.7}$$

由 (2.6) 的右端非负得到下面的关键不等式

$$(u^k - u^*)^T Q(u^k - \tilde{u}^k) \geq \varphi(u^k, \tilde{u}^k), \quad \forall\, u^* \in \Omega^*. \tag{2.8}$$

这一节中的收缩算法, 都是基于向量 $(u^k - \tilde{u}^k)$ 的算法. 然而, 我们并不直接受用 $(u^k - \tilde{u}^k)$, 而是用

$$d(u^k, \tilde{u}^k) = M(u^k - \tilde{u}^k) \tag{2.9}$$

作为寻查方向, 其中

$$M = D^{-1}Q, \tag{2.10}$$

矩阵 $Q$ 由 (2.5) 给出, 矩阵

$$D = \begin{pmatrix} rI_n & 0 \\ 0 & sI_m \end{pmatrix}. \tag{2.11}$$

根据这些定义, 我们有

$$M = \begin{pmatrix} I_n & \frac{1}{r}A^T \\ 0 & I_m \end{pmatrix},$$

是对角部分为单位阵的上三角分块矩阵. 关系式 (2.7) 和 (2.8) 可以写成等价的

$$\varphi(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k). \tag{2.12}$$

和

$$\langle D(u^k - u^*), M(u^k - \tilde{u}^k)\rangle \geq \varphi(u^k, \tilde{u}^k), \quad \forall\, u^* \in \Omega^*. \tag{2.13}$$

## 2.2 初等的收缩算法

The Primary Contraction Methods (初等的收缩算法) 是指用确定的方向取单位步长的收缩算法.

**The Primary Contraction Methods** 对给定的 $u^k$ 和由 (2.1) 生成的 $\tilde{u}^k$, 我们用

$$u^{k+1} = u^k - M(u^k - \tilde{u}^k) \tag{2.14}$$

生成新的迭代点. 在用 (2.1) 生成的 $\tilde{u}^k$ 的过程中, 采用 Armijo 法则, 可以做到



**对给定的 $r > 0$ 和 $\nu \in (0,1)$, 选取 $s$ 使得**

$$\frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2 \leq \nu\big(r\|x^k - \tilde{x}^k\|^2 + s\|\lambda^k - \tilde{\lambda}^k\|^2\big). \tag{2.15}$$

特别当 $s$ 取得使 $sr\nu \geq \|A^TA\|$ 时, 条件 (2.15) 自然成立.

我们考虑 $D$-模下的收缩算法. 根据迭代公式 (2.14) 有

$$\begin{aligned} &\|u^k - u^*\|_D^2 - \|u^{k+1} - u^*\|_D^2 \\ &\quad = \|u^k - u^*\|_D^2 - \|(u^k - u^*) - M(u^k - \tilde{u}^k)\|_D^2 \\ &\quad = 2(u^k - u^*)^T DM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_D^2. \end{aligned}$$

对上式右端的 $(u^k - u^*)^T DM(u^k - \tilde{u}^k)$ 使用 (2.12)–(2.13), 我们有

$$\begin{aligned} &\|u^k - u^*\|_D^2 - \|u^{k+1} - u^*\|_D^2 \\ &\quad \geq 2(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_D^2. \end{aligned} \tag{2.16}$$

注意到 (2.16) 的右端等于

$$\begin{aligned}&2(u^k-\tilde{u}^k)^T DM(u^k-\tilde{u}^k)-\|M(u^k-\tilde{u}^k)\|_D^2\\ =\;& (u^k-\tilde{u}^k)^T\big[M^TD+DM-M^TDM\big](u^k-\tilde{u}^k).\end{aligned} \tag{2.17}$$

利用矩阵恒等式

$$M^TD+DM-M^TDM=D-(M^T-I)D(M-I)$$

和 $D$, $M$ 的结构, 经过简单计算就有

$$\begin{aligned}&M^TD+DM-M^TDM\\ =\;&\begin{pmatrix} rI_n & 0\\ 0 & sI_m\end{pmatrix}-\begin{pmatrix} 0 & 0\\ \frac{1}{r}A & 0\end{pmatrix}\begin{pmatrix} rI_n & 0\\ 0 & sI_m\end{pmatrix}\begin{pmatrix} 0 & \frac{1}{r}A^T\\ 0 & 0\end{pmatrix}\\ =\;&\begin{pmatrix} rI_n & 0\\ 0 & sI_m-\frac{1}{r}AA^T\end{pmatrix}.\end{aligned}$$



因此,

$$\begin{aligned}&2(u^k-\tilde{u}^k)^T DM(u^k-\tilde{u}^k)-\|M(u^k-\tilde{u}^k)\|_D^2\\ =\;& \|u^k-\tilde{u}^k\|_D^2-\frac{1}{r}\|A^T(\lambda^k-\tilde{\lambda}^k)\|^2.\end{aligned} \tag{2.18}$$

在条件 (2.15) 满足的情况下, 从 (2.18) 式得到

$$\begin{aligned}&2(u^k-\tilde{u}^k)^T DM(u^k-\tilde{u}^k)-\|M(u^k-\tilde{u}^k)\|_D^2\\ \ge\;& (1-\nu)\big(r\|x^k-\tilde{x}^k\|^2+s\|\lambda^k-\tilde{\lambda}^k\|^2\big).\end{aligned} \tag{2.19}$$

以 (2.19) 代入 (2.16), 就有下面的定理：

**Theorem 2.1** *在基于 Primal-Dual 松弛 PPA 的收缩算法中, 如果产生预测点时条件 (2.15) 成立, 则由初等收缩算法 (2.14) 生成的序列 $\{u^k=(x^k,\lambda^k)\}$ 满足*

$$\|u^{k+1}-u^*\|_D^2\le\|u^k-u^*\|_D^2-(1-\nu)\|u^k-\tilde{u}^k\|_D^2. \tag{2.20}$$

定理 2.1 是保证初等收缩算法收敛的关键不等式.

## 2.3 一般的收缩算法

一般收缩算法同样用给定的方向 $M(u^k - \tilde{u}^k)$, 但通过计算步长确定下一个迭代点. 在 $D$-模的意义下, 使新的迭代点靠解集尽可能近一些. 在用 (2.1) 生成的 $\tilde{u}^k$ 的过程中, 采用 Armijo 法则, 可以做到

**对给定的 $r > 0$, 选取 $s$ 使得**

$$\begin{aligned} & \frac{1}{s}\|A(x^k - \tilde{x}^k)\|^2 + \frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2 \\ & \quad \le \quad 2\big(r\|x^k - \tilde{x}^k\|^2 + s\|\lambda^k - \tilde{\lambda}^k\|^2\big). \end{aligned} \tag{2.21}$$

特别当 $s$ 取得使 $sr \ge \frac{1}{2}\|A^TA\|$ 时, $\|A^TA\| \le 2rs$,

$$\begin{aligned} & \frac{1}{s}\|A(x^k - \tilde{x}^k)\|^2 + \frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2 \\ & \quad \le \quad \frac{1}{s}\|A^TA\| \cdot \|x^k - \tilde{x}^k\|^2 + \frac{1}{r}\|AA^T\| \cdot \|\lambda^k - \tilde{\lambda}^k\|^2, \end{aligned}$$

条件 (2.21) 自然成立.



利用 $D$ 和 $M$ 的表达式以及 Cauchy-Schwarz 不等式, 我们有

$$\begin{aligned} \varphi(u^k, \tilde{u}^k) &= (u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) \\ &= \|u^k - \tilde{u}^k\|_D^2 + (x^k - \tilde{x}^k)^T A^T(\lambda^k - \tilde{\lambda}^k) \\ &\ge r\|x^k - \tilde{x}^k\|^2 - \frac{1}{2}\|x^k - \tilde{x}^k\| \cdot \|A^T(\lambda^k - \tilde{\lambda}^k)\| \\ &\quad + s\|\lambda^k - \tilde{\lambda}^k\|^2 - \frac{1}{2}\|\lambda^k - \tilde{\lambda}^k\| \cdot \|A(x^k - \tilde{x}^k)\| \\ &\ge r\|x^k - \tilde{x}^k\|^2 - \frac{1}{4}\Big\{r\|x^k - \tilde{x}^k\|^2 + \frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2\Big\} \\ &\quad + s\|\lambda^k - \tilde{\lambda}^k\|^2 - \frac{1}{4}\Big\{s\|\lambda^k - \tilde{\lambda}^k\|^2 + \frac{1}{s}\|A(x^k - \tilde{x}^k)\|^2\Big\} \\ &= \frac{3}{4}\Big\{r\|x^k - \tilde{x}^k\|^2 + s\|\lambda^k - \tilde{\lambda}^k\|^2\Big\} \\ &\quad - \frac{1}{4}\Big\{\frac{1}{s}\|A(x^k - \tilde{x}^k)\|^2 + \frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2\Big\}. \end{aligned}$$

在条件 (2.21) 满足的情况下, 根据上式就有

$$\varphi(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) \ge \frac{1}{4}\|u^k - \tilde{u}^k\|_D^2. \tag{2.22}$$

**一般的收缩算法 I** 对给定的 $u^k$ 和由 (2.1) 生成的 $\tilde{u}^k$, 我们用

$$u(\alpha) = u^k - \alpha M(u^k - \tilde{u}^k) \tag{2.23}$$

产生依赖于步长 $\alpha$ 的迭代点. 对任意给定的 $u^* \in \Omega^*$, 我们将

$$\vartheta(\alpha) = \|u^k - u^*\|_D^2 - \|u(\alpha) - u^*\|_D^2 \tag{2.24}$$

看成是本次迭代的"进步量", 它是步长 $\alpha$ 的函数. 利用 (2.23) 和 (2.24) 中 $\vartheta(\alpha)$ 的定义, 我们有

$$\begin{aligned} \vartheta(\alpha) &= \|u^k - u^*\|_D^2 - \|(u^k - u^*) - \alpha M(u^k - \tilde{u}^k)\|_D^2 \\ &= 2\alpha(u^k - u^*)^T DM(u^k - \tilde{u}^k) - \alpha^2\|M(u^k - \tilde{u}^k)\|_D^2. \end{aligned}$$

对上式右端的 $(u^k - u^*)^T DM(u^k - \tilde{u}^k)$ 使用 (2.12) 和 (2.13), 就有

$$\vartheta(\alpha) \geq q(\alpha),$$

其中

$$q(\alpha) = 2\alpha(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) - \alpha^2\|M(u^k - \tilde{u}^k)\|_D^2. \tag{2.25}$$



同样, 注意到 (2.25) 中的 $q(\alpha)$ 是 $\alpha$ 的二次函数, 它在

$$\alpha_k^* = \frac{(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k)}{\|M(u^k - \tilde{u}^k)\|_D^2} = \frac{(u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k)}{(M(u^k - \tilde{u}^k))^T Q(u^k - \tilde{u}^k)} \tag{2.26}$$

时取得极大值.

**当条件 (2.21) 满足时, 对所有的 $k \geq 0$, 都有 $\alpha^* \geq 1/6$.**

根据 (2.26), 为证明上述结论, 只要证明

$$6(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_D^2 \geq 0. \tag{2.27}$$

由 (2.22), 我们有

$$\begin{aligned} &6(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_D^2 \\ \geq\ & \|u^k - \tilde{u}^k\|_D^2 + 2(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_D^2. \end{aligned}$$

再利用 (2.18), 得到

$$\begin{aligned} &\|u^k - \tilde{u}^k\|_D^2 + 2(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_D^2 \\ =\ & 2\|u^k - \tilde{u}^k\|_D^2 - \frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2. \end{aligned}$$

当条件 (2.21) 满足时上式右端非负, (2.27) 成立, 就有 $\alpha_k^* \ge 1/6$.

我们想要极大化 $\vartheta(\alpha)$ (见 (2.24)), 由于它含有未知 $u^*$, 我们不得已才极大化它的下界函数 $q(\alpha)$. 在实际计算中, 我们取

$$u^{k+1} = u^k - \gamma \alpha_k^* M(u^k - \tilde{u}^k), \tag{2.28}$$

为新的迭代点, 其中 $\gamma \in [1, 2)$ 称为松弛因子.

利用 $\vartheta(\alpha) \ge q(\alpha)$, 将 (2.25) 中的 $\alpha$ 置换成 $\gamma\alpha^*$, 并用 (2.26), 就有

$$\begin{aligned} \|u^{k+1} - u^*\|_D^2 &\le \|u^k - u^*\|_D^2 - q(\gamma\alpha^*) \\ &= \quad \|u^k - u^*\|_D^2 - \gamma(2-\gamma)\alpha_k^*(u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k). \end{aligned}$$

根据上面的不等式, 由 $\alpha_k^* \ge 1/6$ 和关系式 (2.22), 就得到下面的定理

**Theorem 2.2** *在基于Primal-Dual 松弛 PPA 的收缩算法中, 如果产生预测点时条件 (2.21) 成立, 则由一般收缩算法 (2.28) 生成的序列$\{u^k = (x^k, \lambda^k)\}$ 满足*

$$\|u^{k+1} - u^*\|_D^2 \le \|u^k - u^*\|_D^2 - \frac{\gamma(2-\gamma)}{24}\|u^k - \tilde{u}^k\|_D^2. \tag{2.29}$$

迭代序列 $\{u^k\}$ 是 $D$-模下 Fejér 单调的.



**一般的收缩算法 II** 对给定的 $u^k$ 和由 (2.1) 生成的 $\tilde{u}^k$, 我们也可以用

$$u^{k+1} = u^k - \gamma \alpha_k^* Q^{-T} D(u^k - \tilde{u}^k) \tag{2.30}$$

生成新的迭代点, 由于矩阵

$$Q^{-T}D = \begin{pmatrix} \frac{1}{r}I_n & 0 \\ -\frac{1}{rs}A & \frac{1}{s}I_m \end{pmatrix}\begin{pmatrix} rI_n & 0 \\ 0 & sI_m \end{pmatrix} = \begin{pmatrix} I_n & 0 \\ -\frac{1}{s}A & I_m \end{pmatrix}$$

的形式是简单的, 校正 (2.30) 容易实现, 步长则由

$$\alpha_k^* = (u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k)/\|u^k - \tilde{u}^k\|_D^2, \qquad \gamma \in (0, 2)$$

给出. 由 (2.22), 可知 $\alpha_k^* \ge 1/4$. 对 $H = QD^{-1}Q^T$, 利用 (2.12) 和 (2.13), 则有

$$\begin{aligned} \|u^{k+1} - u^*\|_H^2 &= \|u^k - u^* - \gamma\alpha^* Q^{-T} D(u^k - \tilde{u}^k)\|_H^2 \\ &\le \quad \|u^k - u^*\|_H^2 - 2\gamma\alpha_k^*(u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k) + \gamma^2(\alpha_k^*)^2\|u^k - \tilde{u}^k\|_D^2 \\ &= \quad \|u^k - u^*\|_H^2 - \gamma(2-\gamma)\alpha_k^*(u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k). \\ &\le \quad \|u^k - u^*\|_H^2 - \frac{\gamma(2-\gamma)}{16}\|u^k - \tilde{u}^k\|_D^2. \end{aligned}$$

由于 $u^*$ 是任意给定的解点, 这样生成的序列 $\{u^k\}$ 在 $H$-模下 Fejér 单调的.

# 3 Dual-Primal 松弛 PPA 收缩算法

通过对经典PPA (1.10) 中下半部分中的 $\tilde{x}^k$ 松弛成 $x^k$ 构造 Relaxed PPA. 子问题 (1.10) 就简化成求 $(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega$, 使得对任意的 $u = (x, \lambda) \in \Omega$, 都有

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ Ax^k - b \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \geq 0. \tag{3.1}$$

松弛以后的子问题 (3.1) 是容易求解的. 首先对 (3.1) 的下半 (Dual) 部分, 根据变分不等式和投影的关系, $\tilde{\lambda}^k$ 可以通过

$$\tilde{\lambda}^k = P_\Lambda[\lambda^k - \frac{1}{s}(Ax^k - b)]$$

直接给出. 有了 $\tilde{\lambda}^k$, (3.1) 的上半 (Primal) 部分中要求的只剩下 $\tilde{x}^k$, 它可以通过求解极小化问题

$$\min\left\{\theta(x) + \frac{r}{2}\|x - [x^k + \frac{1}{r}A^T\tilde{\lambda}^k]\|^2 \,\middle|\, x \in \mathcal{X}\right\}$$

得到. 根据假设条件, 这是一个容易求解的问题.



用松弛 PPA 按先 $\tilde{\lambda}^k$ (dual) 后 $\tilde{x}^k$ (primal) 的顺序生成预测点 $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k)$, 然后构造的收缩算法, 我们称之为基于Dual-Primal 松弛 PPA 的收缩算法 (Dual - primal relaxed PPA based contraction method).

## 3.1 Dual-Primal 生成预测点

**Dual–Primal Method 生成预测点** 对给定的 $(x^k, \lambda^k)$ 和 $s > 0$, 由

$$\tilde{\lambda}^k = P_\Lambda\{\lambda^k - \frac{1}{s}(Ax^k - b)\} \tag{3.2a}$$

生成 Dual 预测点 $\tilde{\lambda}^k$. 然后选取适当的 $r > 0$ 并求解

$$\min\left\{\theta(x) + \frac{r}{2}\|x - [x^k + \frac{1}{r}A^T\tilde{\lambda}^k]\|^2 \,\middle|\, x \in \mathcal{X}\right\} \tag{3.2b}$$

得到 Primal 预测点 $\tilde{x}^k$ (如何选取 $r > 0$ 放在后面讨论).

根据前面的分析 (见 (3.1)), 由 (3.2) 生成的预测点 $(\tilde{x}^k,\tilde{\lambda}^k)$ 满足

$$(\tilde{x}^k,\tilde{\lambda}^k)\in\Omega,\quad \theta(x)-\theta(\tilde{x}^k)+\begin{pmatrix} x-\tilde{x}^k\\ \lambda-\tilde{\lambda}^k\end{pmatrix}^T\left\{\begin{pmatrix} -A^T\tilde{\lambda}^k\\ A\tilde{x}^k-b\end{pmatrix}\right.$$
$$\left.+\begin{pmatrix} r(\tilde{x}^k-x^k)\\ -A(\tilde{x}^k-x^k)\end{pmatrix}+\begin{pmatrix} 0\\ s(\tilde{\lambda}^k-\lambda^k)\end{pmatrix}\right\}\geq 0,\quad \forall(x,\lambda)\in\Omega.$$

它的紧凑形式是

$$\tilde{u}^k\in\Omega,\ \theta(x)-\theta(\tilde{x}^k)+(u-\tilde{u}^k)^T\{F(\tilde{u}^k)+Q(\tilde{u}^k-u^k)\}\geq 0,\ \forall u\in\Omega, \tag{3.3}$$

其中矩阵(对应于 Dual-Primal 方法生成预测点的 $Q$, 有时也记成 $Q_{DP}$)

$$Q_{DP}=\begin{pmatrix} rI_n & 0\\ -A & sI_m\end{pmatrix}, \tag{3.4}$$

是非对称的. 类似地, 可以得到下面的关键不等式

$$(\tilde{u}^k-u^*)^TQ(u^k-\tilde{u}^k)\geq 0,\quad \forall\, u^*\in\Omega^*. \tag{3.5}$$



采用

$$M(u^k-\tilde{u}^k),\qquad (\text{其中}\quad M=D^{-1}Q) \tag{3.6}$$

作为寻查方向, 其中矩阵 $H$ 如 (2.14) 给出. 根据这些定义, 我们有

$$M(u^k-\tilde{u}^k)=\begin{pmatrix} x^k-\tilde{x}^k\\ \lambda^k-\tilde{\lambda}^k\end{pmatrix}+\begin{pmatrix} 0 & 0\\ -\frac{1}{s}A & 0\end{pmatrix}\begin{pmatrix} x^k-\tilde{x}^k\\ \lambda^k-\tilde{\lambda}^k\end{pmatrix}. \tag{3.7}$$

同时有

$$(u^k-u^*)^TDM(u^k-\tilde{u}^k)\geq\varphi(u^k,\tilde{u}^k),\quad \forall\, u^*\in\Omega^*, \tag{3.8}$$

其中

$$\varphi(u^k,\tilde{u}^k)=(u^k-\tilde{u}^k)^TDM(u^k-\tilde{u}^k). \tag{3.9}$$

## 3.2 初等的收缩算法

这里的收缩算法是一种预测校正方法. 初等的收缩算法 (Primary Contraction Methods) 是在选取确定的方向后, 校正产生新迭代点时取单位步长的算法.

**The Primary Contraction Methods** 对给定的 $u^k$ 和由 (3.2) 生成的 $\tilde{u}^k$, 我们用

$$u^{k+1} = u^k - M(u^k - \tilde{u}^k) \tag{3.10}$$

生成新的迭代点. 在用 (3.2) 生成的 $\tilde{u}^k$ 的过程中, 采用 Armijo 法则, 可以做到

**对给定的$s > 0$ 和 $\nu \in (0,1)$, 选取 $r$ 使得**

$$\frac{1}{s}\|A(x^k - \tilde{x}^k)\|^2 \le \nu\big(r\|x^k - \tilde{x}^k\|^2 + s\|\lambda^k - \tilde{\lambda}^k\|^2\big). \tag{3.11}$$

特别当 $s$ 取得使 $sr\nu \ge \|A^TA\|$ 时, 条件 (3.11) 自然成立.

与定理 2.1 类似, 对基于Dual-Primal 松弛 PPA 的收缩算法, 有如下的定理:

**Theorem 3.1** *在基于Dual-Primal 松弛 PPA 的收缩算法中, 如果产生预测点时条件 (3.11) 成立, 则由初等收缩算法 (3.10) 生成的序列 $\{u^k = (x^k, \lambda^k)\}$ 满足*

$$\|u^{k+1} - u^*\|_D^2 \le \|u^k - u^*\|_D^2 - (1-\nu)\|u^k - \tilde{u}^k\|_D^2. \tag{3.12}$$

定理 3.1 是保证初等收缩算法收敛的关键不等式.



## 3.3 一般的收缩算法

一般收缩算法用同样的给定方向 $M(u^k - \tilde{u}^k)$, 通过计算步长确定下一个迭代点. 在 $H$-模的意义下, 使新的迭代点靠解集尽可能近一些.

**对给定的 $s > 0$, 选取 $r$ 使得**

$$\begin{aligned} &\frac{1}{s}\|A(x^k - \tilde{x}^k)\|^2 + \frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2 \\ \le\ & 2\big(r\|x^k - \tilde{x}^k\|^2 + s\|\lambda^k - \tilde{\lambda}^k\|^2\big). \end{aligned} \tag{3.13}$$

这是能够办到的. 当 $s$ 取得使 $sr \ge \frac{1}{2}\|A^TA\|$ 时, 条件 (3.13) 自然成立.

类似与 (2.21)–(2.22) 的分析, 在条件 (3.13) 满足的情况下, 有

$$\varphi(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k) \ge \frac{1}{4}\|u^k - \tilde{u}^k\|_D^2. \tag{3.14}$$

**一般的收缩算法 I** 对给定的 $u^k$ 和由 (3.2) 生成的 $\tilde{u}^k$, 我们取

$$u^{k+1} = u^k - \gamma\alpha_k^* M(u^k - \tilde{u}^k), \tag{3.15}$$

为新的迭代点, 其中

$$\alpha_k^* = (u^k - \tilde{u}^k)^T DM(u^k - \tilde{u}^k)/\|M(u^k - \tilde{u}^k)\|_D^2 \tag{3.16}$$

$\gamma \in [1,2)$ 称为松弛因子. 同样, 在条件 (3.13) 满足时, 会有 $\alpha_k^* > 1/6$.

**Theorem 3.2** *在基于Dual-Primal 松弛 PPA 的收缩算法中, 如果产生预测点时条件 (3.13) 成立, 则由收缩算法 (3.15) 生成的序列 $\{u^k = (x^k, \lambda^k)\}$ 满足*

$$\|u^{k+1} - u^*\|_D^2 \le \|u^k - u^*\|_D^2 - \frac{\gamma(2-\gamma)}{24}\|u^k - \tilde{u}^k\|_D^2. \tag{3.17}$$

同样, 迭代序列 $\{u^k\}$ 是 $D$-模下 Fejér 单调的, 上式是证明收敛的关键不等式.

**一般的收缩算法 II** 同样, 对给定的 $u^k$ 和由 (3.2) 生成的 $\tilde{u}^k$, 也可以用

$$u^{k+1} = u^k - \gamma\alpha_k^* Q^{-T} D(u^k - \tilde{u}^k) \tag{3.18}$$

生成新的迭代点, 其中 $\alpha_k^* = \varphi(u^k, \tilde{u}^k)/\|u^k - \tilde{u}^k\|_D^2$, 取 $H = QD^{-1}Q^T$, 就有

$$\|u^{k+1} - u^*\|_H^2 \le \|u^k - u^*\|_H^2 - \frac{\gamma(2-\gamma)}{16}\|u^k - \tilde{u}^k\|_D^2.$$

由于 $u^*$ 是任意给定的解点, 这样生成的序列 $\{u^k\}$ 在 $H$-模下 Fejér 单调.



# 4 数值计算

我们以第四讲中试验的相关性矩阵校正和矩阵完整化问题作为例子, 并与前一讲的 PPA 算法比较计算效果. 我们采用基于Dual-Primal 松弛 PPA 的收缩算法.

## 4.1 相关性矩阵校正问题

相关性矩阵校正问题的数学形式是

$$\min\{\frac{1}{2}\|X - C\|_F^2 \mid \operatorname{diag}(X) = e,\ X \in S_+^n\}, \tag{4.1}$$

在第四讲的 §4.1 中已经作了介绍. 用 $z \in \Re^n$ 作为等式约束 $\operatorname{diag}(X) = e$ 的 Lagrange 乘子. 我们用 (3.2) 生成问题 (4.1) 的预测点, 问题 (3.2b) 是求解

$$\min\{\frac{1}{2}\|X - C\|_F^2 + \frac{r}{2}\|X - [X^k + \frac{1}{r}\operatorname{diag}(\tilde{z}^k)]\|_F^2 | X \in S_+^n\}. \tag{4.2}$$

这样的工作在前一讲的 §4.1 作了介绍, 求 $(\tilde{x}^k, \tilde{\lambda}^k)$ 的主要工作是做 EIG 分解.

Relaxed PPA-Primal method 只是将前一讲 Code 4.1 (Classical PPA) 的第 (9) 行的
`A=(X0*r + C + diag(yt*2 -y0))/(1+r)` 改成 `A=(X0*r + C + diag(yt))/(1+r)`
并将 (12) 行的 `y = y0 - EY` 改成 `y = y0 - (EY - diag(EX)/s).`

### Code 4.1. Relaxed PPA – Primal method $rs = 1.01,\ s = 0.5$

```
%%%    RePPA (primal mothed) for calibrating correlation matrix   %(1)
function RePPA_MP(n,C,r,s,tol);                                   %(2)
X=eye(n);      y=zeros(n,1);    tic;      %% The initial iterate %(3)
stopc=1;                         k=0;                             %(4)
while (stopc>tol && k<=100)        %% Beginning of an Iteration  %(5)
 if mod(k,20)==0 fprintf(' k=%4d   epsm=%9.3e  \n',k,stopc); end; %(6)
    X0=X;    y0=y;    k=k+1;                                      %(7)
    yt=y0 - (diag(X0)-ones(n,1))/s;             EY=y0-yt;         %(8)
    A=(X0*r + C + diag(yt))/(1+r);                                %(9)
    [V,D]=eig(A);   D=max(0,D);  XT=(V*D)*V';      EX=X0-XT;     %(10)
    ex=max(max(abs(EX)));  ey=max(abs(EY));   stopc=max(ex,ey);  %(11)
    X=X0 - EX;             y=y0 - (EY - diag(EX)/s;              %(12)
end;                                        % End of an Iteration %(13)
toc;                 TB = max(abs(diag(X-eye(n))));              %(14)
fprintf(' k=%4d  epsm=%9.3e   max|X_jj - 1|=%8.5f \n',k,stopc,TB); %%
```

采用 Relaxed-PPA-Prmal method 与 Classical-PPA method 计算效果完全一样.



我们下面比较 Relaxed PPA – general Method 与前一讲的 Extended PPA Method 的计算效率.

从 Relaxed PPA – Primal Method 到 Relaxed PPA – general Method, 其差别在:

- Code 4.2 与 Code 4.1 的前 11 行完全相同.
- 在 Code 4.2 中, 增加的 12–14 行是为了计算步长.
- Code 4.2 中的第 15 行给出新的迭代点, 与 Code 4.1 中给出新迭代点的第 11 行相比, 寻查方向相同, 只是步长不同. Code 4.1 中用的是单位步长. Code 4.2 中的步长要用公式 (3.16) 计算.
- 为使用 Relaxed PPA-Primal Method, 需要满足条件 $rs \geq \|A^TA\|$ (见 (3.11)), 由于 $\|A^TA\| = 1$, 我们取 $rs = 1.01$, $s = 0.5$.
- 对使用 Relaxed PPA-General Method, 只需要满足条件 $rs \geq \frac{1}{2}\|A^TA\|$ (见 (3.13)), 我们取 $rs = 0.65$, $s = 0.4$. 相对于步长为 $1$ 的算法, 这里生成预测点时, 将参数 $s$ 和 $r$ 都乘上一个因子 $0.8$.

## Code 4.2 Relaxed PPA – general method $rs = 0.65, \; s = 0.4$

```
%%%  RePPA_MG (general mothed)for calibrating correlation matrix  %(1)
function RePPA_MG(n,C,r,s,tol,gamma)                              %(2)
X=eye(n);      y=zeros(n,1);    tic;     %%  The initial iterate  %(3)
stopc=1;                    k=0;                                  %(4)
while (stopc>tol && k<=100)          %% Beginning of an Iteration  %(5)
 if mod(k,20)==0 fprintf(' k=%4d   epsm=%9.3e  \n',k,stopc); end; %(6)
    X0=X;     y0=y;     k=k+1;                                    %(7)
    yt=y0 - (diag(X0)-ones(n,1))/s;                EY=y0-yt;      %(8)
    A=(X0*r + C + diag(yt))/(1+r);                                %(9)
    [V,D]=eig(A);   D=max(0,D);  XT=(V*D)*V';     EX=X0-XT;      %(10)
    ex=max(max(abs(EX)));  ey=max(abs(EY));   stopc=max(ex,ey);  %(11)
    T1=EX(:)'*EX(:);     T2=EY(:)'*EY(:);                        %(12)
    dEX=diag(EX);         T12 =EY'*dEX;       T3=dEX'*dEX;       %(13)
    alpha=(T1*r + T2*s - T12)/(T1*r + T2*s - T12*2 + T3/s);      %(14)
    X=X0-EX*(alpha*gamma);    y=y0-(EY-dEX/s)*(alpha*gamma);     %(15)
end;                                        % End of an Iteration %(16)
toc;                     TB = max(abs(diag(X-eye(n))));          %(17)
fprintf(' k=%4d  epsm=%9.3e   max|X_jj - 1|=%8.5f \n',k,stopc,TB); %%
```



## 矩阵校正问题 (4.1)–使用 Matlab 中的 eig 子程序

| $n \times n$ Matrix | Extended PPA | | Relaxed PPA | |
|---|---|---|---|---|
| $n =$ | No. It | CPU Sec. | No. It | CPU Sec. |
| 100 | 22 | 0.24 | 22 | 0.24 |
| 200 | 25 | 1.42 | 22 | 1.24 |
| 500 | 27 | 11.66 | 22 | 9.62 |
| 800 | 29 | 50.47 | 23 | 39.77 |
| 1000 | 31 | 99.26 | 25 | 78.95 |
| 2000 | 41 | 883.76 | 33 | 713.36 |

♣ 关于相关系数矩阵校正的程序在附件的 Codes-05 的文件夹“*矩阵校正*”的 MaT-EIG 中. 只要运行 demo.m, 输入 n 就可以了.
其中的 PPAC.m 和 PPAG.m 分别是 Classical PPA 和 Extended PPA 的子程序.
RePPAMP.m 和 RePPAMG.m 分别是 Relaxed PPA-Primal Method 和 Relaxed PPA-General Method 的子程序.

♣ 如果改用 Kim TOH 写的 mexeig 做 $[V, D] = \text{mexeig}(A)$, 计算时间大为节省. 相应的程序在附件的 Codes-05 的文件夹"*矩阵校正*" 的 Mex-EIG 中.

矩阵校正问题 (4.1)–特征值分解使用 mexeig

| $n \times n$ Matrix | Extended PPA | | Relaxed PPA | |
|---|---|---|---|---|
| $n =$ | No. It | CPU Sec. | No. It | CPU Sec. |
| 100 | 22 | 0.09 | 22 | 0.09 |
| 200 | 25 | 0.37 | 22 | 0.32 |
| 500 | 27 | 3.35 | 22 | 2.67 |
| 800 | 29 | 11.56 | 23 | 9.12 |
| 1000 | 31 | 22.58 | 25 | 18.20 |
| 2000 | 39 | 209.90 | 33 | 168.79 |

从计算实践看, 对同样的问题, Relaxed PPA-General Method 的计算效果比 Extended PPA 要好. 这两个方法生成预测点的工作量是相同的, 虽然用 Relaxed PPA-General Method 需要计算步长, 因为计算步长是 $O(n^2)$ 的运算, Relaxed PPA-General Method 花费的时间还是有较大幅度的节省.



## 4.2 矩阵完整化方面的应用

矩阵完整化问题的试验例子与前一讲的 §4.2 相同, 算例来自 [1]. 我们只将 Relaxed PPA-General Method 与 Extended PPA 做比较. 这两个方法生成预测点的工作量是相同的.

♣ 矩阵完整化用 Matlab SVD 的程序在附件的 Codes-05 的文件夹 矩阵完整化–SVD-MaT 中. 只要运行 demo.m 就可以了. 要对不同情形试验, 只要在 demo.m 中用 % 做适当选择. 其中的 PPAGMaT.m 和 PPAMMaT.m 分别是 Extended PPA 和 Relaxed PPA 的子程序, 均采用 $\gamma = 1.5$.

如果用 Matlab 中的标准 SVD, 对同样规模的问题, 计算时间与迭代次数成正比. 虽然用 Relaxed PPA-General Method 需要计算步长, 因为计算步长是 $O(n^2)$ 的运算, 这在总的计算量中是微不足道的. 我们将Relaxed PPA-General Method 求解矩阵完整化问题的程序作为 Code 4.3 列在后面.

用 Relaxed PPA-General method 求解矩阵完整化问题与前一讲的 Code 4.3 (Extended PPA) 的差别是：将第 (10) 行的

`A=X0 + (YT*2 -Y0)/r` 改成 `A = X0 + (YT)/r` .

在 Code 4.3 (Extended PPA) 第 (13) 行以后加上三行计算步长 $\alpha = \gamma\alpha^*$.

并将 Code 4.3 (Extended PPA) 的 (15)-(16) 行改成了现在的 (18)-(19) 行.

矩阵完整化问题：用 Matlab 中标准SVD 求解结果

| Unknown $n \times n$ matrix $M$ | | | | Extended PPA | | Relaxed PPA | |
|---|---|---|---|---|---|---|---|
| $n$ | rank$(ra)$ | $m/d_{ra}$ | $m/n^2$ | $\#$iters | times(Sec.) | $\#$iters | times(Sec.) |
| 1000 | 10 | 6 | 0.12 | 77 | 867.82 | 56 | 629.67 |
| 1000 | 50 | 4 | 0.39 | 37 | 411.28 | 27 | 302.27 |
| 1000 | 100 | 3 | 0.58 | 31 | 362.58 | 29 | 305.93 |

因此, 用 Relaxed PPA-General Method 花费的时间有较大幅度的节省.

♣ 注意到, [1] 中的方法对这三个例子的迭代次数分别是 117, 114 和 129 (See the first three examples in Table 5.1 of [1], pp. 1974). 他们的方法每次迭代的主要工作量也是做一次 SVD 分解, 由于采用了不完全分解技术, [1] 中节省了每次 SVD 分解的运行时间. 我们调用的是 Matlab 中标准SVD, 减少了迭代次数, 但没有节省总的运行时间.



## Code 4.3. Relaxed PPA for Matrix Completion Problem

```
function PPAE(n,r,s,M,Omega,maxIt,tol,gamma)    %  Ititial Process %%(1)
X=zeros(n);     Y=zeros(n);      YT=zeros(n);                       %(2)
  nM0=norm(M(Omega),'fro');    eps=1;  VioKKT=1;  k=0;   tic;       %(3)
%% Minimum nuclear norm solution by PPA method                      %(4)
while  (eps > tol && k<= maxIt)                                     %(5)
 if mod(k,5)==0                                                     %(6)
 fprintf('It=%3d |X-M|/|M|=%9.2e VioKKT=%9.2e\n',k,eps,VioKKT); end;%(7)
  k=k+1;      X0=X;      Y0=Y;                                      %(8)
  YT(Omega)=Y0(Omega)-(X0(Omega)-M(Omega))/s;            EY=Y-YT;   %(9)
  A = X0 + (YT)/r;                       [U,D,V]=svd(A,0);         %(10)
  D=D-eye(n)/r;     D=max(D,0);     XT=(U*D)*V';         EX=X-XT;  %(11)
  DXM=XT(Omega)-M(Omega);           eps = norm(DXM,'fro')/nM0;     %(12)
  VioKKT = max( max(max(abs(EX)))*r, max(max(abs(EY))) );          %(13)
  T1=EX(:)'*EX(:);       T2=EY(:)'*EY(:);                          %(14)
  T12 = EY(:)'*EX(:);    EXOm = EX(Omega);  T3 = EXOm(:)'*EXOm(:); %(15)
  alpha =(T1*r + T2*s - T12)*gamma/(T1*r + T2*s - T12*2 + T3/s);   %(16)
   if (eps <= tol)       alpha=1; end;                             %(17)
  X = X0 - EX*alpha;                                               %(18)
  Y(Omega) = Y0(Omega) -  (EY(Omega) - EXOm/s)*alpha;              %(19)
end                                                                %(20)
 fprintf('It=%3d  |X-M|/|M|=%9.2e  Vi0KKT=%9.2e \n',k,eps,VioKKT); %(21)
   RelEr=norm((X-M),'fro')/norm(M,'fro');   toc;                   %(22)
 fprintf(' Relative error = %9.2e  Rank(X)=%3d \n',RelEr,rank(X)); %(23)
 fprintf(' Violation of KKT Condition = %9.2e  \n',VioKKT);        %(24)
```

## 4.3 Min-Max 问题上的应用

对第一讲 §4.2 中提到用全变差极小处理图像去模糊 [2] 的 min-max 问题,

$$\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} \ \Phi(x,y) := \theta_1(x) + y^TAx - \theta_2(y). \tag{4.3}$$

它可以转换成等价的变分不等式: 求 $(x^*,y^*)\in\mathcal{X}\times\mathcal{Y}$, 使得

$$\begin{pmatrix} x-x^* \\ y-y^* \end{pmatrix}^T \begin{pmatrix} f(x^*)+A^Ty^* \\ g(y^*)-Ax^* \end{pmatrix} \ge 0, \quad \forall (x,y)\in\mathcal{X}\times\mathcal{Y}, \tag{4.4}$$

其中 $f(x)\in\partial\theta_1(x)$, $g(y)\in\partial\theta_2(y)$. 用这一讲介绍的Relaxed PPA 方法去求解, 对给定的 $(x^k,y^k)$, 可以通过

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + y^TAx^k + \frac{s}{2}\|y-y^k\|^2 \,|\, y\in\mathcal{Y}\}. \tag{4.5a}$$

$$\tilde{x}^k = \text{Argmin}\Big\{\theta_1(x) + \frac{r}{2}\big\|x - \big[x^k - \frac{1}{r}A^T\tilde{y}^k\big]\big\|^2 \,\big|\, x\in\mathcal{X}\Big\}. \tag{4.5b}$$

或者



$$\tilde{x}^k = \text{Argmin}\{\theta_1(x) + x^TA^Ty^k + \frac{r}{2}\|x-x^k\|^2 \,|\, x\in\mathcal{X}\}. \tag{4.6a}$$

$$\tilde{y}^k = \text{Argmin}\Big\{\theta_2(x) + \frac{s}{2}\big\|y - \big[y^k + \frac{1}{s}A\tilde{x}^k\big]\big\|^2 \,\big|\, y\in\mathcal{Y}\Big\}. \tag{4.6b}$$

求 $(\tilde{x}^k,\tilde{y}^k)\in\mathcal{X}\times\mathcal{Y}$, 对一切 求 $(x,y)\in\mathcal{X}\times\mathcal{Y}$, 都有

$$\begin{pmatrix} x-\tilde{x}^k \\ y-\tilde{y}^k \end{pmatrix}^T \left\{ \begin{pmatrix} f(\tilde{x}^k)+A^T\tilde{y}^k \\ g(\tilde{y}^k)-A\tilde{x}^k \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k-x^k) \\ +A(\tilde{x}^k-x^k)+s(\tilde{y}^k-y^k) \end{pmatrix} \right\} \ge 0. \tag{4.7}$$

由 (4.6) 生成的 $(\tilde{x}^k,\tilde{y}^k)\in\mathcal{X}\times\mathcal{Y}$, 对一切 求 $(x,y)\in\mathcal{X}\times\mathcal{Y}$, 都有

$$\begin{pmatrix} x-\tilde{x}^k \\ y-\tilde{y}^k \end{pmatrix}^T \left\{ \begin{pmatrix} f(\tilde{x}^k)+A^T\tilde{y}^k \\ g(\tilde{y}^k)-A\tilde{x}^k \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k-x^k)-A^T(\tilde{y}^k-y^k) \\ s(\tilde{y}^k-y^k) \end{pmatrix} \right\} \ge 0. \tag{4.8}$$

无论是 (4.7) 还是 (4.8), 都可以写成

$$\tilde{u}^k\in\Omega, \qquad (u-\tilde{u}^k)^T\{F(\tilde{u}^k) + Q(\tilde{u}^k-u^k)\} \ge 0, \quad \forall\, u\in\Omega, \tag{4.9}$$

的形式, 所不同的只是

$$Q = \begin{pmatrix} rI_n & 0 \\ A & sI_m \end{pmatrix} \quad \text{和} \quad Q = \begin{pmatrix} rI_n & -A^T \\ 0 & sI_m \end{pmatrix}.$$

都是当 $rs > \frac{1}{4}\|A^TA\|$ 时, 矩阵 $Q^T + Q$ 正定. 我们建议取 $rs \approx 0.7\|A^TA\|$. 对由 (2.11) 定义的矩阵 $D$, 新的迭代点$u^{k+1}$ 用迭代式

$$u^{k+1} = u^k - \gamma\alpha_k^* M(u^k - \tilde{u}^k), \quad \gamma \in [1, 2),$$

生成(一般取 $\gamma = 1.5$). 采用

$$M = D^{-1}Q, \qquad \alpha_k^* = \frac{(u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k)}{\|M(u^k - \tilde{u}^k)\|_D^2}, \tag{4.10}$$

生成的序列 $\{u^k\}$ 在 $D$-模下 Fejér 单调; 若采用

$$M = Q^{-T}D, \qquad \alpha_k^* = \frac{(u^k - \tilde{u}^k)^T Q(u^k - \tilde{u}^k)}{\|u^k - \tilde{u}^k\|_D^2}, \tag{4.11}$$

生成的序列 $\{u^k\}$ 则在 $H$-模下 Fejér 单调的($H = QD^{-1}Q^T$). 类似的结论可参阅本讲的 §2.3 和 §3.3. 用这类方法求解图像去模糊问题可以参考 [8].



# References


[1] J. F. Cai, E. J. Candès and Z. W. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim., **20**, 1956-1982, 2010.

[2] A. Chambolle and T. Pock, A first-order primal-dual algorithms for convex problem with applications to imaging, J. Math. Imaging Vison, **40**, 120-145, 2011.

[3] B.S. He, A new method for a class of linear variational inequalities, Math. Progr., **66**, 137-144, 1994.

[4] B.S. He, A class of projection and contraction methods for monotone variational inequalities, Applied Mathematics and Optimization, **35**, 69-76, 1997.

[5] B.S. He, Inexact implicit methods for monotone general variational inequalities, Math. Program. Series A, **86**, 199-217, 1999

[6] B. S. He, X. L. Fu and Z.K. Jiang, Proximal point algorithm using a linear proximal term, JOTA **141**, 209-239, 2009.

[7] B. S. He and M.-H. Xu, A general framework of contraction methods for monotone variational inequalities, Pacific J. Optimization, **4**, 195-212, 2008.

[8] B.S. He and X. M. Yuan, Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective, SIAM Journal on Imaging Science, **5**, 119-149, 2012.

[9] R. M. Larsen, Propack – software for large and sparse svd calculation, http://sun.stanford.edu/rmunk/PROPACK/, 2005.

[10] B. Martinet, Regularisation, d'inéquations variationelles par approximations succesives, Rev. Francaise d'Inform. Recherche Oper., **4**, 154-159, 1970.

[11] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Control Optim., **14**, 877-898, 1976.

# 凸优化和单调变分不等式的收缩算法

## 第六讲: 线性约束凸优化扩展问题的定制 PPA 和松弛 PPA 收缩算法

PPA-based contraction methods for general linearly constrained convex optimization

南京大学数学系　　何炳生
hebma@nju.edu.cn



# 1　线性约束凸优化扩展问题

我们考虑如下的线性约束凸优化扩展问题:

$$\min\{\theta(x) \mid Ax \in \mathcal{B},\ x \in \mathcal{X}\} \tag{1.1}$$

其中 $\theta(x)$ 是凸函数, $A \in \Re^{m\times n}$, $\mathcal{B} \subset \Re^m$, $\mathcal{X} \subset \Re^n$ 为闭凸集. 如果 $\mathcal{B} = \{b\}$, 问题 (1.1) 就是前两讲讨论的问题. 推广前两讲的方法, 用来求解问题(1.1).

对任意给定的 $r > 0$ 和 $a \in \Re^n$, 通篇我们假设

**子问题** $\min\{\theta(x) + \frac{r}{2}\|x - a\|^2 \mid x \in \mathcal{X}\}$ **的求解是简单的.**

除此以外, 我们还假设

**在闭凸集 $\mathcal{B}$ 上的投影是容易实现的.**

显然, 分裂可行性问题 [8, 10],

$$\text{Find a point } \ x \in \mathcal{X} \ \text{ such that } \ Ax \in \mathcal{B},$$

是 (1.1) 中置 $\theta(x) \equiv 0$ 的一种特殊形式.

## 线性约束凸优化扩展问题等价的变分不等式

引进辅助变量 $y$. 问题 (1.1) 可以改写成

$$\min\{\theta(x) \mid Ax - y = 0,\ x \in \mathcal{X},\ y \in \mathcal{B}\}. \tag{1.2}$$

对线性约束 $Ax - y = 0$ 引入 Lagrange 乘子 $\lambda \in \Re^m$, 问题 (1.2) 的 Lagrange 函数是定义在 $\mathcal{X} \times \mathcal{B} \times \Re^m$ 上的

$$L(x, y, \lambda) = \theta(x) - \lambda^T (Ax - y).$$

求解问题 (1.1) 就相当于找一组 $(x^*, y^*, \lambda^*)$, 使得

$$\left\{\begin{array}{lll} x^* \in \mathcal{X}, & \theta(x) - \theta(x^*) + (x - x^*)^T(-A^T\lambda^*) \geq 0, & \forall\, x \in \mathcal{X}, \\ y^* \in \mathcal{B}, & (y - y^*)^T \lambda^* \geq 0, & \forall\, y \in \mathcal{B}, \\ \lambda^* \in \Re^m, & (\lambda - \lambda^*)^T (Ax^* - y^*) \geq 0, & \forall\, \lambda \in \Re^m. \end{array}\right. \tag{1.3}$$



通过定义

$$w = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, \qquad F(w) = \begin{pmatrix} -A^T\lambda \\ \lambda \\ Ax - y \end{pmatrix} \tag{1.4}$$

和

$$\mathcal{W} = \mathcal{X} \times \mathcal{B} \times \Re^m,$$

变分不等式 (1.3) 能够写成

$$\text{VI}(\mathcal{W}, F, \theta) \quad w^* \in \mathcal{W},\ \theta(x) - \theta(x^*) + (w - w^*)^T F(w^*) \geq 0,\ \forall\, w \in \mathcal{W}. \tag{1.5}$$

注意到仿射算子 $F$ 是单调的. 由于 $y$ 是辅助变量, 我们记

$$u = \begin{pmatrix} x \\ \lambda \end{pmatrix} \qquad \text{和} \qquad \Omega = \mathcal{X} \times \Re^m.$$

设 $\mathcal{W}^*$ 是 (1.5) 的解集, 并令

$$\Omega^* = \{(x^*, \lambda^*) \mid (x^*, y^*, \lambda^*) \in \mathcal{W}^*\}.$$

我们分别给出为求解线性约束凸优化扩展问题 (1.1) 的 PPA 收缩算法 和 Relaxed PPA 收缩算法.

# 2 线性约束凸优化扩展问题的 PPA 算法

## 2.1 变分不等式框架下 Primal-Dual PPA 算法

对给定的 $u^k=(x^k,\lambda^k)$, 假设我们能给出 $\tilde{w}^k=(\tilde{x}^k,\tilde{y}^k,\tilde{\lambda}^k)$, 使得

$$(\tilde{x}^k,\tilde{y}^k,\tilde{\lambda}^k)\in\mathcal{W},\quad \theta(x)-\theta(\tilde{x}^k)+\begin{pmatrix} x-\tilde{x}^k\\ y-\tilde{y}^k\\ \lambda-\tilde{\lambda}^k\end{pmatrix}^T\left\{\begin{pmatrix} -A^T\tilde{\lambda}^k\\ \tilde{\lambda}^k\\ A\tilde{x}^k-\tilde{y}^k\end{pmatrix}\right.$$

$$\left.+\begin{pmatrix} r(\tilde{x}^k-x^k)+A^T(\tilde{\lambda}^k-\lambda^k)\\ 0\\ A(\tilde{x}^k-x^k)+s(\tilde{\lambda}^k-\lambda^k)\end{pmatrix}\right\}\geq 0,\quad \forall (x,y,\lambda)\in\mathcal{W}. \tag{2.1}$$



将任意的 $w^*=(x^*,y^*,\lambda^*)$ 代入上式中的 $(x,y,\lambda)\in\mathcal{W}$, 并利用 $F(w)$ 的表达式 (1.4), 就有

$$\begin{aligned}&\begin{pmatrix}\tilde{x}^k-x^*\\ \tilde{\lambda}^k-\lambda^*\end{pmatrix}^T\left\{\begin{pmatrix} r(x^k-\tilde{x}^k)\\ A(x^k-\tilde{x}^k)\end{pmatrix}+\begin{pmatrix} A^T(\lambda^k-\tilde{\lambda}^k)\\ s(\lambda^k-\tilde{\lambda}^k)\end{pmatrix}\right\}\\ &\geq\quad \theta(\tilde{x}^k)-\theta(x^*)+(\tilde{w}^k-w^*)^TF(\tilde{w}^k).\end{aligned} \tag{2.2}$$

注意到

$$(\tilde{w}^k-w^*)^TF(\tilde{w}^k)=(\tilde{w}^k-w^*)^TF(w^*).$$

对 $\tilde{w}^k\in\mathcal{W}$, 有

$$\theta(\tilde{x}^k)-\theta(x^*)+(\tilde{w}^k-w^*)^TF(w^*)\geq 0.$$

因此 (2.2) 的右端非负, 我们得到

$$(\tilde{u}^k-u^*)^TG(u^k-\tilde{u}^k)\geq 0, \tag{2.3}$$

其中

$$G=\begin{pmatrix} rI_n & A^T\\ A & sI_m\end{pmatrix}.$$

由于 $rs > \|A^TA\|$ 时, 矩阵 $G$ 是对称正定的. 根据 (2.3), 我们有

$$(u^k - u^*)^T G(u^k - \tilde{u}^k) \geq \|u^k - \tilde{u}^k\|_G^2, \quad \forall\, u^* \in \Omega^*. \tag{2.4}$$

我们用

$$u^{k+1} = u^k - \gamma(u^k - \tilde{u}^k), \quad \gamma \in (0,2) \tag{2.5}$$

生成新的迭代点 $u^{k+1}$, 一般取 $\gamma \in [1.2, 1.9]$.

**Theorem 2.1** *The sequence $\{u^k = (x^k, \lambda^k)\}$ generated by the proposed PPA method satisfies*

$$\|u^{k+1} - u^*\|_G^2 \leq \|u^k - u^*\|_G^2 - \gamma(2-\gamma)\|u^k - \tilde{u}^k\|_G^2. \tag{2.6}$$

- B.S. He and X. M. Yuan, A contraction method with implementable proximal regularization for linearly constrained convex programming, Optimization Online, 2010.

**对给定的 $(x^k, \lambda^k)$, 如何求得 (2.1) 的解 $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \mathcal{W}$**

注意到 (2.1) 第一部分是求 $\tilde{x}^k \in \mathcal{X}$, 使得

$$\theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T\{r(\tilde{x}^k - x^k) - A^T\lambda^k\} \geq 0, \quad \forall x \in \mathcal{X}.$$



因此,

$$\tilde{x}^k = \text{Arg}\min\Big\{\theta(x) + \frac{r}{2}\big\|x - \big[x^k + \frac{1}{r}A^T\lambda^k\big]\big\|^2 \,\Big|\, x \in \mathcal{X}\Big\}.$$

有了 $\tilde{x}^k$, 注意到 (2.1) 的中间与第三部分在一起是

$$\begin{cases} \tilde{y}^k \in \mathcal{B}, \quad (y - \tilde{y}^k)^T\tilde{\lambda}^k \geq 0, \quad \forall y \in \mathcal{B} \\ \big[A(2\tilde{x}^k - x^k) - \tilde{y}^k\big] + s(\tilde{\lambda}^k - \lambda^k) = 0. \end{cases} \tag{2.7}$$

对任意的 $s > 0$, 根据变分不等式与投影方程的关系, (2.7) 的第一部分有

$$\tilde{y}^k = P_{\mathcal{B}}[\tilde{y}^k - s\tilde{\lambda}^k]. \tag{2.8}$$

而 (2.7) 的下半部分可以改写成

$$\tilde{y}^k = \big(A(2\tilde{x}^k - x^k) - s\lambda^k\big) + s\tilde{\lambda}^k. \tag{2.9}$$

将 (2.9) 代入 (2.8) 的右端, 就有

$$\tilde{y}^k = P_{\mathcal{B}}[A(2\tilde{x}^k - x^k) - s\lambda^k].$$

以上式中的 $\tilde{y}^k$ 代入 (2.9), 就得到一个消去了 $y$ 的 $\tilde{\lambda}^k$ 的表达式.

$$\tilde{\lambda}^k = \frac{1}{s}\Big\{P_{\mathcal{B}}[A(2\tilde{x}^k - x^k) - s\lambda^k] - [A(2\tilde{x}^k - x^k) - s\lambda^k]\Big\}.$$

用 PPA 算法求解凸优化问题 (1.1), 总假设 $r, s > 0$ 满足 $rs > \|A^T A\|$. 在导出算法的时候我们引进了辅助变量 $y$, 在算法实现的时候又利用 (2.8) 和(2.9) 消去辅助变量 $y$, 等到如下的只有 $x$ 和 $\lambda$ 参加迭代的公式.

**Primal-Dual 生成 PPA 算法预测点** 对给定的 $(x^k, \lambda^k)$, 先用

$$\tilde{x}^k = \arg\min\{\theta(x) + \frac{r}{2}\|x - [x^k + \frac{1}{r}A^T\lambda^k]\|^2 \,|\, x \in \mathcal{X}\}. \tag{2.10a}$$

求出预测点的 Primal 部分 $\tilde{x}^k$. 然后令

$$\tilde{\lambda}^k = \frac{1}{s}\Big\{P_{\mathcal{B}}[A(2\tilde{x}^k - x^k) - s\lambda^k] - [A(2\tilde{x}^k - x^k) - s\lambda^k]\Big\} \tag{2.10b}$$

得到预测点的 Dual 部分 $\tilde{\lambda}^k$.



## 2.2 变分不等式框架下 Dual-Primal PPA 算法

与 (2.1) 类似, 对给定的 $u^k = (x^k, \lambda^k)$, 也可考虑给出 $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)$, 使得

$$(\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \mathcal{W}, \quad \theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{\begin{pmatrix} -A^T\tilde{\lambda}^k \\ \tilde{\lambda}^k \\ A\tilde{x}^k - \tilde{y}^k \end{pmatrix}\right.$$

$$\left. + \begin{pmatrix} r(\tilde{x}^k - x^k) - A^T(\tilde{\lambda}^k - \lambda^k) \\ 0 \\ -A(\tilde{x}^k - x^k) + s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix}\right\} \geq 0, \quad \forall (x, y, \lambda) \in \mathcal{W}. \tag{2.11}$$

同样的分析也会得到

$$(\tilde{u}^k - u^*)^T G(u^k - \tilde{u}^k) \geq 0.$$

所不同的是, 这里的 $G = \begin{pmatrix} rI_n & -A^T \\ -A & sI_m \end{pmatrix}$. 这时要先解 (2.11) 中只含 $\tilde{y}^k, \tilde{\lambda}^k$

的部分, 采用求解 (2.7) 的方法求解如下的方程组.

$$\begin{cases} \tilde{y}^k \in \mathcal{B}, \quad (y-\tilde{y}^k)^T\tilde{\lambda}^k \geq 0, \quad \forall y \in \mathcal{B} \\ (Ax^k - \tilde{y}^k) + s(\tilde{\lambda}^k - \lambda^k) = 0. \end{cases} \tag{2.12}$$

**Dual-Primal 生成 PPA 算法预测点** 对给定的 $u^k = (x^k, \lambda^k)$, 先用

$$\tilde{\lambda}^k = \frac{1}{s}\Big\{P_{\mathcal{B}}[Ax^k - s\lambda^k] - (Ax^k - s\lambda^k)\Big\} \tag{2.13a}$$

给出预测点的 Dual 部分 $\tilde{\lambda}^k$. 再通过求解

$$\min\big\{\theta(x) + \frac{r}{2}\big\|x - [x^k + \frac{1}{r}A^T(2\tilde{\lambda}^k - \lambda^k)]\big\|^2 \,\big|\, x \in \mathcal{X}\big\}. \tag{2.13b}$$

得到预测点的 Primal 部分 $\tilde{x}^k$.

的确, 在第一节的假设条件下, 用 (2.10) 或者 (2.13) 求得 $(\tilde{x}^k, \tilde{\lambda}^k)$ 是容易实现的. 若用

$$u^{k+1} = u^k - \gamma(u^k - \tilde{u}^k), \quad \gamma \in (0, 2)$$

生成新的迭代点, 这些方法是第四讲 **Customized PPA** 方法的直接推广.



# 3 基于Primal-Dual 松弛 PPA 的收缩算法

我们考虑用 Relaxed PPA 求解扩展问题 (1.1) , 还是将 $y$ 看作辅助变量. 求解变分不等式 (1.3)-(1.4) 的经典的 PPA 算法是从 $u^k = (x^k, \lambda^k)$ 出发, 求 $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \mathcal{W}$, 使得

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ \tilde{\lambda}^k \\ A\tilde{x}^k - \tilde{y}^k \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) \\ 0 \\ s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \geq 0, \ \forall w \in \mathcal{W}. \tag{3.1}$$

直接求解 (3.1) 显然是有困难的.

**Relaxed PPA** 考虑将第一部分中的 $\tilde{\lambda}^k$ 松弛成 $\lambda^k$, 或者将第三部分中的 $\tilde{x}^k$ 松弛成 $x^k$, 求解松弛后的问题, 生成预测点, 构造收缩算法.

## 3.1 Primal-Dual 生成预测点

如果我们将 (3.1) 第一部分中的 $\tilde{\lambda}^k$ 换 (松弛) 成 $\lambda^k$, 就得到一个 Relaxed PPA 子问题: 求 $\tilde{w}^k=(\tilde{x}^k,\tilde{y}^k,\tilde{\lambda}^k)\in\mathcal{W}$, 使得

$$\theta(x)-\theta(\tilde{x}^k)+\begin{pmatrix}x-\tilde{x}^k\\ y-\tilde{y}^k\\ \lambda-\tilde{\lambda}^k\end{pmatrix}^T\left\{\begin{pmatrix}-A^T\lambda^k\\ \tilde{\lambda}^k\\ A\tilde{x}^k-\tilde{y}^k\end{pmatrix}+\begin{pmatrix}r(\tilde{x}^k-x^k)\\ 0\\ s(\tilde{\lambda}^k-\lambda^k)\end{pmatrix}\right\}\geq 0,\ \forall w\in\mathcal{W}. \tag{3.2}$$

注意到 (3.2) 可以写成

$$(\tilde{x}^k,\tilde{y}^k,\tilde{\lambda}^k)\in\mathcal{W},\quad \theta(x)-\theta(\tilde{x}^k)+\begin{pmatrix}x-\tilde{x}^k\\ y-\tilde{y}^k\\ \lambda-\tilde{\lambda}^k\end{pmatrix}^T\left\{\begin{pmatrix}-A^T\tilde{\lambda}^k\\ \tilde{\lambda}^k\\ A\tilde{x}^k-\tilde{y}^k\end{pmatrix}\right.$$
$$\left.+\begin{pmatrix}r(\tilde{x}^k-x^k)-A^T(\tilde{\lambda}^k-\lambda^k)\\ 0\\ s(\tilde{\lambda}^k-\lambda^k)\end{pmatrix}\right\}\geq 0,\quad \forall(x,y,\lambda)\in\mathcal{W}.$$



将上式的 $(x,y,\lambda)\in\mathcal{W}$ 用任意的 $w^*=(x^*,y^*,\lambda^*)$ 代替, 并利用 $F(w)$ 的表达式 (1.4), 就有

$$\begin{aligned}&\begin{pmatrix}\tilde{x}^k-x^*\\ \tilde{\lambda}^k-\lambda^*\end{pmatrix}^T\left\{\begin{pmatrix}r(x^k-\tilde{x}^k)\\ 0\end{pmatrix}+\begin{pmatrix}A^T(\lambda^k-\tilde{\lambda}^k)\\ s(\lambda^k-\tilde{\lambda}^k)\end{pmatrix}\right\}\\ \geq\quad&\theta(\tilde{x}^k)-\theta(x^*)+(\tilde{w}^k-w^*)^TF(\tilde{w}^k).\end{aligned} \tag{3.3}$$

注意到 (3.3) 的右端非负. 因此有

$$(\tilde{u}^k-u^*)^TQ(u^k-\tilde{u}^k)\geq 0,\quad \forall\, u^*\in\Omega^*, \tag{3.4}$$

其中

$$Q=\begin{pmatrix}rI_n & A^T\\ 0 & sI_m\end{pmatrix} \tag{3.5}$$

是非对称的. 根据 (3.4), 我们有

$$(u^k-u^*)^TQ(u^k-\tilde{u}^k)\geq(u^k-\tilde{u}^k)^TQ(u^k-\tilde{u}^k),\ \ \forall\, u^*\in\Omega^*. \tag{3.6}$$

基于 Relaxed PPA 的收缩算法, 我们都考虑 $H$-模下的收缩, 其中

$$H = \begin{pmatrix} rI_n & 0 \\ 0 & sI_m \end{pmatrix}, \tag{3.7}$$

是正定的分块数量矩阵. 我们都采用

$$d(u^k, \tilde{u}^k) = M(u^k - \tilde{u}^k) \tag{3.8}$$

作为寻查方向, 其中

$$M = H^{-1}Q, \qquad M = \begin{pmatrix} I_n & \frac{1}{r}A^T \\ 0 & I_m \end{pmatrix} \tag{3.9}$$

是分块上三角矩阵, 其对角部分是单位矩阵. 注意到关系式 (3.6) 则为

$$(u^k - u^*)^T HM(u^k - \tilde{u}^k) \ge (u^k - \tilde{u}^k)^T HM(u^k - \tilde{u}^k), \quad \forall\, u^* \in \Omega^*, \tag{3.10}$$

**对给定的 $(x^k, \lambda^k)$, 如何求得 (3.2) 的解 $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \mathcal{W}$**



注意到 (3.2) 最上面的部分是

$$\tilde{x}^k \in \mathcal{X}, \ \theta(x') - \theta(\tilde{x}^k) + (x' - \tilde{x}^k)^T\{-A^T\lambda^k + r(\tilde{x}^k - x^k)\} \ge 0, \ \forall x' \in \mathcal{X}.$$

未知的只有 $\tilde{x}^k$, 它可以通过求解子问题

$$\min\left\{\theta(x) + \frac{r}{2}\left\|x - \left[x^k + \frac{1}{r}A^T\lambda^k\right]\right\|^2 \,\middle|\, x \in \mathcal{X}\right\} \tag{3.11}$$

得到. 由 $\mathcal{W} = \mathcal{X} \times \mathcal{B} \times \Re^m$, 根据变分不等式和投影方程之间的关系, (3.2) 的中间部分可以写成

$$\tilde{y}^k = P_{\mathcal{B}}[\tilde{y}^k - s\tilde{\lambda}^k], \tag{3.12}$$

其中 $s$ 为任意大于零的常数. 根据 (3.2) 的第三部分, 有

$$\tilde{y}^k = A\tilde{x}^k + s(\tilde{\lambda}^k - \lambda^k). \tag{3.13}$$

以 (3.13) 中的 $\tilde{y}^k$ 代入(3.12), 就得到由 $\tilde{x}^k$ 和 $\lambda^k$ 给出的 $\tilde{\lambda}^k$ 的表达式

$$\tilde{\lambda}^k = \frac{1}{s}\left\{P_{\mathcal{B}}[A\tilde{x}^k - s\lambda^k] - [A\tilde{x}^k - s\lambda^k]\right\}. \tag{3.14}$$

根据上面的分析, 我们得到了生成预测点 $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k)$ 的方法.

**Primal–Dual Method 生成预测点** 对给定的 $(x^k, \lambda^k)$ 和 $r > 0$, 通过求解

$$\min\left\{\theta(x) + \frac{r}{2}\left\|x - \left[x^k + \frac{1}{r}A^T\lambda^k\right]\right\|^2 \,\middle|\, x \in \mathcal{X}\right\} \tag{3.15a}$$

得到 Primal 预测点 $\tilde{x}^k$. 再选取适当的 $s > 0$ 并用

$$\tilde{\lambda}^k = \frac{1}{s}\left\{P_{\mathcal{B}}[A\tilde{x}^k - s\lambda^k] - [A\tilde{x}^k - s\lambda^k]\right\} \tag{3.15b}$$

生成 Dual 预测点 $\tilde{\lambda}^k$ (如何选取 $s > 0$ 放在后面讨论).

把用松弛 PPA 按先 $\tilde{x}^k$ (primal) 后 $\tilde{\lambda}^k$ (dual) 的顺序生成预测点 $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k)$, 然后构造的收缩算法称为基于Primal-Dual 松弛 PPA 的收缩算法 (Primal-dual relaxed PPA based contraction method).

## 3.2 初等的收缩算法

The Primary Contraction Methods (初等的收缩算法) 是指对确定的方向取单位步长的收缩算法.



**The Primary Contraction Methods** 对给定的 $u^k$ 和由 (3.15) 生成的 $\tilde{u}^k$, 用

$$u^{k+1} = u^k - M(u^k - \tilde{u}^k) \tag{3.16}$$

生成新的迭代点. 在由 (3.15) 生成的 $\tilde{u}^k$ 的过程中, 采用 Armijo 法则, 要求做到

**对给定的 $r > 0$ 和 $\nu \in (0,1)$, 选取 $s$ 使得**

$$\frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2 \le \nu\big(r\|x^k - \tilde{x}^k\|^2 + s\|\lambda^k - \tilde{\lambda}^k\|^2\big). \tag{3.17}$$

特别当 $s$ 取得使 $sr\nu \ge \|A^TA\|$ 时, 条件 (3.17) 自然成立.

根据迭代公式 (3.16) 有

$$\begin{aligned}
&\|u^k - u^*\|_H^2 - \|u^{k+1} - u^*\|_H^2 \\
&\quad = \|u^k - u^*\|_H^2 - \|(u^k - u^*) - M(u^k - \tilde{u}^k)\|_H^2 \\
&\quad = 2(u^k - u^*)^T HM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_H^2.
\end{aligned}$$

对上式右端的 $(u^k-u^*)^T HM(u^k-\tilde{u}^k)$ 使用 (3.10), 我们有

$$
\begin{aligned}
&\|u^k-u^*\|_H^2-\|u^{k+1}-u^*\|_H^2 \\
&\quad\geq\quad 2(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)-\|M(u^k-\tilde{u}^k)\|_H^2. \qquad (3.18)
\end{aligned}
$$

对 (3.18) 的右端进一步处理得

$$
\begin{aligned}
&2(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)-\|M(u^k-\tilde{u}^k)\|_H^2 \\
&\quad=\quad (u^k-\tilde{u}^k)^T(HM+M^TH-M^THM)(u^k-\tilde{u}^k).
\end{aligned}
$$

利用矩阵恒等式

$$
M^TH+HM-M^THM=H-(M^T-I)H(M-I)
$$

和 $H$ 与 $M$ 的定义 (见 (3.7) 和 (3.9)), 经过简单计算就有



$$
\begin{aligned}
&M^TH+HM-M^THM \\
&=\quad \begin{pmatrix} rI_n & 0 \\ 0 & sI_m \end{pmatrix}-\begin{pmatrix} 0 & 0 \\ \frac{1}{r}A & 0 \end{pmatrix}\begin{pmatrix} rI_n & 0 \\ 0 & sI_m \end{pmatrix}\begin{pmatrix} 0 & \frac{1}{r}A^T \\ 0 & 0 \end{pmatrix} \\
&=\quad \begin{pmatrix} rI_n & 0 \\ 0 & sI_m-\frac{1}{r}AA^T \end{pmatrix}.
\end{aligned}
$$

因此得到

$$
\begin{aligned}
&2(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)-\|M(u^k-\tilde{u}^k)\|_H^2 \\
&\quad=\quad \|u^k-\tilde{u}^k\|_H^2-\frac{1}{r}\|A^T(\lambda^k-\tilde{\lambda}^k)\|^2. \qquad (3.19)
\end{aligned}
$$

在条件 (3.17) 满足的情况下, 从 (3.19) 式得到

$$
\begin{aligned}
&2(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)-\|M(u^k-\tilde{u}^k)\|_H^2 \\
&\quad\geq\quad (1-\nu)\big(r\|x^k-\tilde{x}^k\|^2+s\|\lambda^k-\tilde{\lambda}^k\|^2\big). \qquad (3.20)
\end{aligned}
$$

以 (3.20) 代入 (3.18), 就有下面的定理：

**Theorem 3.1** *在基于 Primal-Dual 松弛 PPA 的收缩算法中, 如果产生预测点时条件 (3.17) 成立, 则由初等收缩算法 (3.16) 生成的序列 $\{u^k = (x^k, \lambda^k)\}$ 满足*

$$\|u^{k+1} - u^*\|_H^2 \le \|u^k - u^*\|_H^2 - (1-\nu)\|u^k - \tilde{u}^k\|_H^2. \tag{3.21}$$

定理 3.1 是保证初等收缩算法收敛的关键不等式.

## 3.3 一般的收缩算法

一般收缩算法用同样的给定方向 $d(u^k, \tilde{u}^k)$, 通过计算步长确定下一个迭代点. 在 $H$-模的意义下, 使新的迭代点靠解集尽可能近一些.

**一般的收缩算法 I** 对给定的 $u^k$ 和由 (3.15) 生成的 $\tilde{u}^k$, 用我们用

$$u(\alpha) = u^k - \alpha M(u^k - \tilde{u}^k) \tag{3.22}$$

产生依赖于步长 $\alpha$ 的迭代点. 对由 (3.15) 生成的 $\tilde{u}^k$, 要求做到



**对给定的 $r > 0$, 选取 $s$ 使得**

$$\begin{aligned}&\frac{1}{s}\|A(x^k - \tilde{x}^k)\|^2 + \frac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2 \\ &\quad \le \quad 2\big(r\|x^k - \tilde{x}^k\|^2 + s\|\lambda^k - \tilde{\lambda}^k\|^2\big).\end{aligned} \tag{3.23}$$

特别当 $s$ 取得使 $sr \ge \frac{1}{2}\|A^TA\|$ 时, 条件 (3.23) 自然成立.

利用 $H$ 和 $M$ 的表达式以及 Cauchy-Schwarz 不等式, 我们有

$$\begin{aligned}
&(u^k - \tilde{u}^k)^T HM(u^k - \tilde{u}^k) \\
&\quad = \quad \|u^k - \tilde{u}^k\|_H^2 + (x^k - \tilde{x}^k)^T A^T(\lambda^k - \tilde{\lambda}^k) \\
&\quad \ge \quad r\|x^k - \tilde{x}^k\|^2 - \tfrac{1}{2}\|x^k - \tilde{x}^k\| \cdot \|A^T(\lambda^k - \tilde{\lambda}^k)\| \\
&\qquad\quad +s\|\lambda^k - \tilde{\lambda}^k\|^2 - \tfrac{1}{2}\|\lambda^k - \tilde{\lambda}^k\| \cdot \|A(x^k - \tilde{x}^k)\| \\
&\quad \ge \quad r\|x^k - \tilde{x}^k\|^2 - \tfrac{1}{4}\big\{r\|x^k - \tilde{x}^k\|^2 + \tfrac{1}{r}\|A^T(\lambda^k - \tilde{\lambda}^k)\|^2\big\} \\
&\qquad\quad +s\|\lambda^k - \tilde{\lambda}^k\|^2 - \tfrac{1}{4}\big\{s\|\lambda^k - \tilde{\lambda}^k\|^2 + \tfrac{1}{s}\|A(x^k - \tilde{x}^k)\|^2\big\}.
\end{aligned}$$

因此,

$$
\begin{aligned}
&(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)\\
\ge\;& \frac{3}{4}\Big\{r\|x^k-\tilde{x}^k\|^2+s\|\lambda^k-\tilde{\lambda}^k\|^2\Big\}\\
&-\frac{1}{4}\Big\{\frac{1}{s}\|A(x^k-\tilde{x}^k)\|^2+\frac{1}{r}\|A^T(\lambda^k-\tilde{\lambda}^k)\|^2\Big\}.
\end{aligned}
$$

在条件 (3.23) 满足的情况下, 根据上式就有

$$
(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)\ge\frac{1}{4}\|u^k-\tilde{u}^k\|_H^2. \tag{3.24}
$$

对任意给定的 $u^*\in\Omega^*$, 我们将

$$
\vartheta(\alpha):=\|u^k-u^*\|_H^2-\|u(\alpha)-u^*\|_H^2 \tag{3.25}
$$

看成是本次迭代的 “进步量”, 它是步长 $\alpha$ 的函数. 利用 (3.22) 和 (3.25) 中 $\vartheta(\alpha)$ 的定义, 我们有

$$
\begin{aligned}
\vartheta(\alpha) &= \|u^k-u^*\|_H^2-\|(u^k-u^*)-\alpha M(u^k-\tilde{u}^k)\|_H^2\\
&= 2\alpha(u^k-u^*)^T HM(u^k-\tilde{u}^k)-\alpha^2\|M(u^k-\tilde{u}^k)\|_H^2.
\end{aligned}
$$



对上式右端的 $(u^k-u^*)^T HM(u^k-\tilde{u}^k)$ 使用 (3.10) 就有

$$
\vartheta(\alpha)\ge q(\alpha),
$$

其中

$$
q(\alpha)=2\alpha(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)-\alpha^2\|M(u^k-\tilde{u}^k)\|_H^2. \tag{3.26}
$$

同样, 注意到 (3.26) 中的 $q(\alpha)$ 是 $\alpha$ 的二次函数, 它在

$$
\alpha_k^*=\frac{(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)}{\|M(u^k-\tilde{u}^k)\|_H^2} \tag{3.27}
$$

时取得极大值.

**当条件 (3.23) 满足时, 对所有的 $k\ge 0$, 都有 $\alpha^*\ge 1/6$.**

根据 (3.27), 为证明上述结论, 只要证明

$$
6(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)-\|M(u^k-\tilde{u}^k)\|_H^2\ge 0. \tag{3.28}
$$

由 (3.24), 我们有

$$
\begin{aligned}
& 6(u^k-\tilde u^k)^T HM(u^k-\tilde u^k) - \|M(u^k-\tilde u^k)\|_H^2 \\
& \quad \geq \quad \|u^k-\tilde u^k\|_H^2 + 2(u^k-\tilde u^k)^T HM(u^k-\tilde u^k) - \|M(u^k-\tilde u^k)\|_H^2.
\end{aligned}
$$

再利用 (3.19), 得到

$$
\begin{aligned}
& \|u^k-\tilde u^k\|_H^2 + 2(u^k-\tilde u^k)^T HM(u^k-\tilde u^k) - \|M(u^k-\tilde u^k)\|_H^2 \\
& \quad = \quad 2\|u^k-\tilde u^k\|_H^2 - \frac{1}{r}\|A^T(\lambda^k-\tilde\lambda^k)\|^2.
\end{aligned}
$$

当条件 (3.23) 满足时上式右端非负, (3.28) 成立, 就有 $\alpha_k^* \geq 1/6$.

我们想要极大化 $\vartheta(\alpha)$ (见 (3.25)), 由于它含有未知 $u^*$, 我们不得已才极大化它的下界函数 $q(\alpha)$ (见 (3.26)).　在实际计算中, 我们取

$$
u^{k+1} = u^k - \gamma\alpha_k^* M(u^k-\tilde u^k), \tag{3.29}
$$

为新的迭代点, 其中 $\gamma \in [1,2)$ 称为松弛因子.

利用 $\vartheta(\alpha) \geq q(\alpha)$, 将 (3.26) 中的 $\alpha$ 置换成 $\gamma\alpha^*$, 并利用 (3.27), 就有

$$
\|u^{k+1}-u^*\|_H^2 \leq \|u^k-u^*\|_H^2 - \gamma(2-\gamma)\alpha_k^*(u^k-\tilde u^k)^T HM(u^k-\tilde u^k). \tag{3.30}
$$



根据 (3.24) 和 (3.30), 以及 $\alpha_k^* \geq 1/6$, 我们得到下面的定理

**Theorem 3.2** *在基于Primal-Dual 松弛 PPA 的收缩算法中, 如果产生预测点时条件 (3.23) 成立, 则由一般收缩算法 (3.29) 生成的序列 $\{u^k = (x^k, \lambda^k)\}$ 满足*

$$
\|u^{k+1}-u^*\|_H^2 \leq \|u^k-u^*\|_H^2 - \frac{\gamma(2-\gamma)}{24}\|u^k-\tilde u^k\|_H^2. \tag{3.31}
$$

迭代序列 $\{u^k\}$ 是Fejér 单调的.

**一般的收缩算法 II**　同样, 对给定的 $u^k$ 和由 (3.15) 生成的 $\tilde u^k$, 也可以用

$$
u^{k+1} = u^k - \gamma\alpha_k^* H^{-1}M^{-T}H(u^k-\tilde u^k) \tag{3.32}
$$

生成新的迭代点, 其中

$$
\alpha_k^* = (u^k-\tilde u^k)^T HM(u^k-\tilde u^k)/\|u^k-\tilde u^k\|_H^2,
$$

由 (3.24) $\alpha_k^* \geq 1/4$. 取 $G = HMH^{-1}M^TH$, 由 (3.10), $\{u^k\}$ 满足收缩性质

$$
\|u^{k+1}-u^*\|_G^2 \leq \|u^k-u^*\|_G^2 - \frac{\gamma(2-\gamma)}{16}\|u^k-\tilde u^k\|_H^2. \tag{3.33}
$$

利用不等式 (3.31) 和 (3.33), 容易证明这些算法的总体收敛性.

# 4 基于Dual-Primal 松弛 PPA 的收缩算法

经典的 PPA 算法是从 $u^k=(x^k,\lambda^k)$ 出发, 求 $\tilde{w}^k=(\tilde{x}^k,\tilde{y}^k,\tilde{\lambda}^k)\in\mathcal{W}$, 使得

$$\theta(x)-\theta(\tilde{x}^k)+\begin{pmatrix} x-\tilde{x}^k\\ y-\tilde{y}^k\\ \lambda-\tilde{\lambda}^k\end{pmatrix}^T\left\{\begin{pmatrix} -A^T\tilde{\lambda}^k\\ \tilde{\lambda}^k\\ A\tilde{x}^k-\tilde{y}^k\end{pmatrix}+\begin{pmatrix} r(\tilde{x}^k-x^k)\\ 0\\ s(\tilde{\lambda}^k-\lambda^k)\end{pmatrix}\right\}\ge 0,\ \forall w\in\mathcal{W}. \tag{4.1}$$

## 4.1 Dual-Primal 生成预测点

**Dual-Primal Relaxed PPA** 是将经典的 PPA 子问题 (4.1) 第三部分中的 $\tilde{x}^k$ 换 (松弛) 成 $x^k$, 就得到如下的一个 Relaxed PPA 子问题: $\tilde{w}^k\in\mathcal{W}$,

$$\theta(x)-\theta(\tilde{x}^k)+\begin{pmatrix} x-\tilde{x}^k\\ y-\tilde{y}^k\\ \lambda-\tilde{\lambda}^k\end{pmatrix}^T\left\{\begin{pmatrix} -A^T\tilde{\lambda}^k\\ \tilde{\lambda}^k\\ Ax^k-\tilde{y}^k\end{pmatrix}+\begin{pmatrix} r(\tilde{x}^k-x^k)\\ 0\\ s(\tilde{\lambda}^k-\lambda^k)\end{pmatrix}\right\}\ge 0,\forall w\in\mathcal{W}. \tag{4.2}$$



如同 §3 中, 将上式的 $w=(x,y,\lambda)\in\mathcal{W}$ 用任意的 $w^*=(x^*,y^*,\lambda^*)$ 代替, 并利用 $F(w)$ 的表达式 (1.4), 就有

$$\begin{aligned}&\begin{pmatrix}\tilde{x}^k-x^*\\ \tilde{\lambda}^k-\lambda^*\end{pmatrix}^T\left\{\begin{pmatrix} r(x^k-\tilde{x}^k)\\ -A(x^k-\tilde{x}^k)\end{pmatrix}+\begin{pmatrix}0\\ s(\lambda^k-\tilde{\lambda}^k)\end{pmatrix}\right\}\\ &\ge\ (\tilde{w}^k-w^*)^TF(\tilde{w}^k)+\theta(\tilde{x}^k)-\theta(x^*).\end{aligned} \tag{4.3}$$

注意到 (4.3) 的右端非负. 因此有

$$(\tilde{u}^k-u^*)^TQ(u^k-\tilde{u}^k)\ge 0,\quad\forall u^*\in\Omega^*, \tag{4.4}$$

其中

$$Q=\begin{pmatrix} rI_n & 0\\ -A & sI_m\end{pmatrix} \tag{4.5}$$

是非对称的. 根据 (4.4), 我们有

$$(u^k-u^*)^TQ(u^k-\tilde{u}^k)\ge(u^k-\tilde{u}^k)^TQ(u^k-\tilde{u}^k)\quad\forall u^*\in\Omega^*. \tag{4.6}$$

这一节中的收缩算法, 我们也采用

$$d(u^k, \tilde{u}^k) = M(u^k - \tilde{u}^k) \tag{4.7}$$

作为寻查方向, 其中 $M = H^{-1}Q$, 矩阵 $H$ 如 (3.7) 给出. 根据这些定义, 我们有

$$M = \begin{pmatrix} I_n & 0 \\ -\frac{1}{s}A & I_m \end{pmatrix}, \tag{4.8}$$

关系式 (4.6) 则为

$$(u^k - u^*)^T HM(u^k - \tilde{u}^k) \ge (u^k - \tilde{u}^k)^T HM(u^k - \tilde{u}^k), \quad \forall\, u^* \in \Omega^*. \tag{4.9}$$

**对给定的 $(x^k, \lambda^k)$, 如何求得 (4.2) 的解 $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \mathcal{W}$**

我们考虑 (4.2) 的中间和第三部分. 注意到 $\mathcal{W} = \mathcal{X} \times \mathcal{B} \times \Re^m$, 根据变分不等式和投影方程之间的关系, (4.2) 的中间部分可以写成

$$\tilde{y}^k = P_{\mathcal{B}}[\tilde{y}^k - \beta\tilde{\lambda}^k], \tag{4.10}$$



其中 $\beta$ 为任意大于零的常数. 根据 (4.2) 的第三部分, 有

$$\tilde{y}^k = Ax^k + s(\tilde{\lambda}^k - \lambda^k). \tag{4.11}$$

以 (4.11) 中的 $\tilde{y}^k$ 代入(4.10), 并令其中的任意大于零的常数 $\beta = s$, 就得到

$$Ax^k + s(\tilde{\lambda}^k - \lambda^k) = P_{\mathcal{B}}[Ax^k + s(\tilde{\lambda}^k - \lambda^k) - s\tilde{\lambda}^k].$$

这样就得到由 $u^k = (x^k, \lambda^k)$ 给出的 $\tilde{\lambda}^k$ 的表达式

$$\tilde{\lambda}^k = \frac{1}{s}\Big\{P_{\mathcal{B}}[Ax^k - s\lambda^k] - [Ax^k - s\lambda^k]\Big\}. \tag{4.12}$$

有了 $\tilde{\lambda}^k$, (4.2) 的第一式中未知的部分只有 $\tilde{x}^k$. 满足

$$\tilde{x}^k \in \mathcal{X},\ \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T\{-A^T\tilde{\lambda}^k + r(\tilde{x}^k - x^k)\} \ge 0,\ \forall\, x \in \mathcal{X}$$

的 $\tilde{x}^k$ 可以通过求解子问题

$$\min\Big\{\theta(x) + \frac{r}{2}\big\|x - \big[x^k + \frac{1}{r}A^T\tilde{\lambda}^k\big]\big\|^2 \,\Big|\, x \in \mathcal{X}\Big\} \tag{4.13}$$

得到. 据此, 我们得到了按先 $\tilde{\lambda}^k$ (dual) 后 $\tilde{x}^k$ (primal) 的顺序生成预测点 $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k)$的方法. 为构造收缩算法走出了关键的一步.

对给定的 $(x^k, \lambda^k)$ 和 $s > 0$, 由

$$\tilde{\lambda}^k = \frac{1}{s}\{P_{\mathcal{B}}[Ax^k - s\lambda^k] - [Ax^k - s\lambda^k]\} \tag{4.14a}$$

给出 Dual 预测点 $\tilde{\lambda}^k$. 然后, 选取适当的 $r > 0$, 通过求解

$$\min\Big\{\theta(x) + \frac{r}{2}\big\|x - \big[x^k + \frac{1}{r}A^T\tilde{\lambda}^k\big]\big\|^2 \,\big|\, x \in \mathcal{X}\Big\} \tag{4.14b}$$

得到 Primal 预测点 $\tilde{x}^k$.

用松弛 PPA 按先 $\tilde{\lambda}^k$ (dual) 后 $\tilde{x}^k$ (primal) 的顺序生成预测点 $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k)$, 然后构造的收缩算法, 称为基于Dual–Primal 松弛 PPA 的收缩算法 (Dual-primal relaxed PPA based contraction method).

## 4.2 初等的收缩算法

The Primary Contraction Methods (初等的收缩算法) 是指对确定的方向取单位步长的收缩算法.



**The Primary Contraction Methods** 对给定的 $u^k$ 和由 (4.14) 生成的 $\tilde{u}^k$, 用

$$u^{k+1} = u^k - M(u^k - \tilde{u}^k) \tag{4.15}$$

生成新的迭代点. 在由 (4.14) 生成的 $\tilde{u}^k$ 的过程中, 采用 Armijo 法则, 可以做到

**对给定的$s > 0$ 和 $\nu \in (0,1)$, 选取 $r$ 使得**

$$\frac{1}{s}\|A(x^k - \tilde{x}^k)\|^2 \le \nu\big(r\|x^k - \tilde{x}^k\|^2 + s\|\lambda^k - \tilde{\lambda}^k\|^2\big). \tag{4.16}$$

特别当 $s$ 取得使 $sr\nu \ge \|A^TA\|$ 时, 条件 (4.16) 自然成立.

**Theorem 4.1** *在基于Dual-Primal 松弛 PPA 的收缩算法中, 如果产生预测点时条件* (4.16) *成立, 则由初等收缩算法* (4.15) *生成的序列* $\{u^k = (x^k, \lambda^k)\}$ *满足*

$$\|u^{k+1} - u^*\|_H^2 \le \|u^k - u^*\|_H^2 - (1-\nu)\|u^k - \tilde{u}^k\|_H^2. \tag{4.17}$$

定理 4.1 是保证初等收缩算法收敛的关键不等式.

## 4.3 一般的收缩算法

一般收缩算法用同样的给定方向 $d(u^k, \tilde{u}^k)$, 通过计算步长确定下一个迭代点. 在 $H$-模的意义下, 使新的迭代点靠解集尽可能近一些. 在由 (4.14) 生成的 $\tilde{u}^k$ 的过程中, 采用 Armijo 法则, 可以做到

**对给定的 $s > 0$, 选取 $r$ 使得**

$$\frac{1}{s}\|A(x^k-\tilde{x}^k)\|^2+\frac{1}{r}\|A^T(\lambda^k-\tilde{\lambda}^k)\|^2 \le 2\big(r\|x^k-\tilde{x}^k\|^2+s\|\lambda^k-\tilde{\lambda}^k\|^2\big). \quad (4.18)$$

特别当 $s$ 取得使 $sr \ge \frac{1}{2}\|A^TA\|$ 时, 条件 (4.18) 自然成立. 类似地, 在条件 (4.18) 满足的情况下, 有

$$\varphi(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k)^T HM(u^k - \tilde{u}^k) \ge \frac{1}{4}\|u^k - \tilde{u}^k\|_H^2. \quad (4.19)$$

**一般的收缩算法 I** 对给定的 $u^k$ 和由 (4.14) 生成的 $\tilde{u}^k$, 取

$$u^{k+1} = u^k - \gamma\alpha_k^* M(u^k - \tilde{u}^k), \quad (4.20)$$

为新的迭代点, 其中

$$\alpha_k^* = (u^k - \tilde{u}^k)^T HM(u^k - \tilde{u}^k)/\|M(u^k - \tilde{u}^k)\|_H^2,$$



$\gamma \in [1, 2)$ 称为松弛因子. 如同在这一讲的 §3.3 中指出的那样, 当条件 (4.18) 满足时, 对所有的 $k \ge 0$, 都有 $\alpha^* \ge 1/6$. 因此有

**Theorem 4.2** *在基于Dual-Primal 松弛 PPA 的收缩算法中, 如果产生预测点时条件 (4.18) 成立, 则由收缩算法 (4.20) 生成的序列$\{u^k = (x^k, \lambda^k)\}$ 满足*

$$\|u^{k+1} - u^*\|_H^2 \le \|u^k - u^*\|_H^2 - \frac{\gamma(2-\gamma)}{24}\|u^k - \tilde{u}^k\|_H^2. \quad (4.21)$$

**一般的收缩算法 II** 同样, 对给定的 $u^k$ 和由 (4.14) 生成的 $\tilde{u}^k$, 也可以用

$$u^{k+1} = u^k - \gamma\alpha_k^* H^{-1}M^{-T}H(u^k - \tilde{u}^k) \quad (4.22)$$

生成新的迭代点, 其中 $\gamma \in (0, 2)$,

$$\alpha_k^* = (u^k - \tilde{u}^k)^T HM(u^k - \tilde{u}^k)/\|u^k - \tilde{u}^k\|_H^2.$$

由 (4.19) $\alpha_k^* \ge 1/4$. 取 $G = HMH^{-1}M^TH$, 根据 (4.9), 序列 $\{u^k\}$ 有收缩性质

$$\|u^{k+1} - u^*\|_G^2 \le \|u^k - u^*\|_G^2 - \frac{\gamma(2-\gamma)}{16}\|u^k - \tilde{u}^k\|_H^2. \quad (4.23)$$

利用不等式 (4.21) 和 (4.23), 容易证明这些算法的总体收敛性.

# 5 应用和计算结果

对第四讲应用一节提出的问题, 讨论相应的扩展问题.

## 5.1 矩阵逼近问题

给定实对称矩阵 $C$, 求某个矩阵集合中的 $X$, 使得 $\|X-C\|_F$ 最小. 我们要求 $X$ 的特征值在一定范围之内, 这可以用

$$X \in S^n_\Lambda = \{H \in \mathcal{S}^n \mid \lambda_{\min} I \preceq H \preceq \lambda_{\max} I\}$$

表示. 另外, 我们要求矩阵 $X$ 的元素在一定范围之内

$$X \in S_B = \{H \in \mathcal{S}^n \mid H_L \le H \le H_U\}.$$

矩阵逼近问题

$$\min_X \{\frac{1}{2}\|X-C\|_F^2 \mid X \in S_B, X \in S^n_\Lambda\} \tag{5.1}$$

是形如 (1.1) 的线性约束凸优化扩展问题并有 $\|A^TA\| = 1$.

**数值试验** 为生成试验例子, 只要给定对称矩阵 $C$.

C=rand(n,n); C=(C'+C)-ones(n,n) + eye(n)

$H_U$=ones(n,n)*0.2; $H_L$=$-H_U$; $H_L(j,j) = H_U(j,j) = 1.$



这样的矩阵 $C$ 的对角元在 $(0,2)$ 之间, 非对角元在 $(-1,1)$ 之间. $S_B$ 就是要求 $X$ 的对角元要等于 1, 非对角元在 $[-0.2,0.2]$ 之间. 我们列出 Extended PPA 算法与 Relaxed PPA 算法的程序.

Matlab Code of the extended Customized PPA

```
%%%     Extended PPA for calibrating correlation matrix            %(1)
function PPA_G(n,C,r,s,HL,HU,tol,gamma)                            %(2)
X=eye(n);      Y=zeros(n);      tic;                               %(3)
fprintf('Extended PPA gamma=1.5 n=%4d \n',n);                      %(4)
stopc=1;                      k=0;                                 %(5)
while (stopc>tol && k<=100)           %% Beginning of an Iteration %(6)
  if mod(k,5)==0  fprintf(' k=%4d epsm=%9.3e \n',k,stopc); end;    %(7)
  X0=X;    Y0=Y;    k=k+1;                                         %(8)
  A= X0-Y0*s;   YT=(min(max(A,HL),HU) - A)/s;       EY=Y0-YT;      %(9)
  A= (X0*r + C + (YT*2-Y0))/(1+r);                                %(10)
  [V,D]=eig(A);  D=max(0,D);  XT=(V*D)*V';          EX=X0-XT;     %(11)
  ex=max(max(abs(EX)));  ey=max(max(abs(EY)));  stopc=max(ex,ey);%(12)
  X=X0 - (X0-XT)*gamma;     Y=Y0 - (Y0-YT)*gamma;                 %(13)
end;                                        %% End of an Iteration %(14)
toc;          TB = max(abs(diag(X-eye(n))));                      %(15)
fprintf('k=%4d epsm=%9.3e  max(abs(X_jj-1))=%8.5f\n',k,stopc,TB);%(16)
```

对同一问题的 Classical PPA 只要置 上述 Customized PPA 程序 第 13 行中的 gamma=1.

## The Matlab Code of the Relaxed PPA

```
%%%      Relaxed PPA for calibrating correlation matrix          %(1)
function PPA_G(n,C,r,s,HL,HU,tol,gamma)                          %(2)
X=eye(n);      Y=zeros(n);      tic;                             %(3)
fprintf('Extended PPA gamma=1.5 n=%4d \n',n);                    %(4)
stopc=1;                      k=0;                               %(5)
while (stopc>tol && k<=100)          %% Beginning of an Iteration %(6)
  if mod(k,5)==0  fprintf(' k=%4d epsm=%9.3e \n',k,stopc); end;   %(7)
  X0=X;    Y0=Y;    k=k+1;                                        %(8)
  A= X0-Y0*s;   YT=(min(max(A,HL),HU) - A)/s;       EY=Y0-YT;     %(9)
  A= (X0*r + C + YT)/(1+r);                                      %(10)
  [V,D]=eig(A);  D=max(0,D);  XT=(V*D)*V';          EX=X0-XT;    %(11)
  ex=max(max(abs(EX)));  ey=max(max(abs(EY)));  stopc=max(ex,ey);%(12)
   T1=EX(:)'*EX(:);   T12=EY(:)'*EX(:);    T2=EY(:)'*EY(:);      %(13)
   alpha=(T1*r + T2*s - T12)*gamma/(T1*r + T2*s - T12*2 + T1/s); %(14)
  X=X0-EX*alpha;      Y=Y0-(EY-EX/s)*alpha;                      %(15)
end;                                      %% End of an Iteration %(16)
toc;          TB = max(abs(diag(X-eye(n))));                     %(17)
fprintf('k=%4d epsm=%9.3e  max(abs(X_jj-1))=%8.5f\n',k,stopc,TB);%(18)
fprintf('    \n');  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

与 Extended PPA 的相比, 采用 Relaxed PPA, 前12行完全相同. 加了(13), (14) 行计算步长. Relaxed PPA 的第(15)行与Extended PPA的第(13)行不同, 因为它们用不同的下降方向.



对带上下界约束的相关性矩阵校正问题, 要做一个在 “BoX” 上的投影, 我们在第 (9) 行中用 YT=(min(max(A,HL),HU) - A)/s 就能实现. 我们将相关性矩阵校正计算用不同方法的计算结果列在下表. 这些方法每步迭代的主要工作量是做一个对称矩阵的 *“特征值-特征向量”* 分解. 对同一种方法, 使用 MexEig 与使用 Matlab 中的 Eig 子程序, 迭代次数相同, 但使用新加坡国立大学Kim TOH 写的 MexEig 做 [V,D]=mexeig(A), 总的花费时间节省不少.

### 相关性矩阵校正计算结果　Matlab EIG　与　MexEig

| $C^{n\times n}$ | Classical PPA | | | Extended PPA | | | Relaxed PPA | | |
|---|---|---|---|---|---|---|---|---|---|
| $n=$ | No. It | Matlab CPU S. | MexEig CPU S. | No. It | Matlab CPU S. | MexEig CPU S. | No. It | Matlab CPU S. | MexEig CPU S. |
| 500 | 47 | 21.02 | 6.36 | 30 | 14.26 | 4.14 | 27 | 12.61 | 3.90 |
| 800 | 49 | 87.80 | 20.80 | 32 | 57.52 | 13.97 | 27 | 49.21 | 12.38 |
| 1000 | 50 | 162.16 | 38.78 | 33 | 107.86 | 26.22 | 28 | 93.75 | 23.10 |
| 1500 | 53 | 525.15 | 127.37 | 37 | 369.09 | 90.69 | 29 | 293.30 | 72.76 |
| 2000 | 57 | 1277.34 | 310.44 | 40 | 901.34 | 220.66 | 30 | 683.60 | 169.23 |

计算结果表明, Extended PPA 比 Classical PPA 快. 此外, 用 Relaxed PPA 比 Extended PPA 更快一些, 虽然用 Relaxed PPA 需要计算步长, 但总的花费时间还是节省不少.

## 5.2 矩阵完整化问题

设 $\Omega$ 是矩阵的下标集. 在矩阵完整化问题的很多情形下, 我们并不要求 $X_\Omega = M_\Omega$, 而是要求 $\|X_\Omega - M_\Omega\|_F \le \delta$. 这时矩阵完整化模型是

$$\min\{\|X\|_* \mid \|X_\Omega - M_\Omega\|_F \le \delta,\ X \in \Re^{m\times n}\} \tag{5.2}$$

这是形如 (1.1) 的线性约束凸优化扩展问题并有 $\|A^TA\| = 1$. 采用本讲介绍的方法, 需要做一个到 "球" 上的投影, 这是容易实现的.

**数值试验** 试验例子是在第四讲 §4.2 中矩阵完整化的例子修改生成的, 它同样来自参考文献 [1]. 生成算例的方法与 [1] 的 Table 5.3 相同, 对 $M$ 中下标属于 $\Pi$ 的元素增加一个满足 $N(0,\sigma^2)$ 分布的噪音。

$$B_{ij} = M_{ij} + Z_{ij},\ Z_{ij} \sim N(0,\sigma^2),\ (ij) \in \Pi$$

其中$\sigma$是已经设定的参数, 并且我们根据经验设定 $\delta = \frac{\|Z_\Pi\|_F}{10}$. 如同 [1] 中, 数值试验的停机准则采取包括噪音的绝对误差 $\|X_\Pi^k - B_\Pi\|_F \le \sqrt{m}\sigma$.

拿使用 Extended PPA 和 Relaxed PPA 的计算结果和文献 [1] 中 Table 5.3 所列的结果比较, 在迭代次数和耗费时间上都有优势的.



我们用下面的程序生成试验问题: 通过在第10 行选 Sigma= 0.01, 0.1 允许不同的噪音

Creating the test examples of Noise Matrix Completion

```
clear all;      clc;                                                  %(1)
maxIt=100;      gamma=1.5;      sigma = 1;  %% choose 0.01,0.1        %(2)
   n=200;     ra=10;      oversampling = 5;  %% Test A                %(3)
%  n=1000;    ra=10;      oversampling = 6;  %% Test 1                %(4)
%  n=1000;    ra=50;      oversampling = 4;  %% Test 2                %(5)
%  n=1000;    ra=100;     oversampling = 3;  %% Test 3                %(6)
%% Generating the test problem                                        %(7)
rd = randseed;         randn('state',rd);                             %(8)
M=randn(n,ra)*randn(ra,n);   %% The matrix will be completed          %(9)
Z=sigma*randn(n,n);          %% The noise matrix                     %(10)
df =ra*(n*2-ra);             %% The freedom of the matrix            %(11)
mo=oversampling;                                                     %(12)
m=min(oversampling*df,round(.99*n*n));%% No. of the known            %(13)
Omega= randsample(n^2,m);              %% Subset Omega               %(14)
B=M;     B(Omega)=M(Omega)+Z(Omega);                                 %(15)
ratio=norm(Z(Omega),'fro')/norm(M(Omega),'fro');                     %(16)
fprintf('Matrix: n=%4d Rank(M)=%3d Oversampling=%2d \n',n,ra,mo);%(17)
fprintf('ratio=%5d\n',ratio);     %% the initial noise ratio         %(18)
fprintf('\n'); M0=norm(M(Omega),'fro');                              %(19)
delta=norm(Z(Omega),'fro')/10; tol=sqrt(m)*sigma/M0;                 %(20)
```

## Extended PPA for Noise Matrix Completion

```
function PPAE(n,r,s,M,Omega,maxIt,gamma,delta,tol)                  %(1)
X=zeros(n); Y=zeros(n); YT=zeros(n); A=zeros(n);                    %(2)
nM0=norm(M(Omega),’fro’); eps=100000; VioKKT=1; k=0;    tic;        %(3)
%%Minimum nuclear norm solution by Extended PPA method              %(4)
while (eps > tol && k<= maxIt)                                      %(5)
if mod(k,10)==0                                                     %(6)
 fprintf(’It=%3d |X-M|/|M|=%9.2eVioKKT=%9.2e\n’,k,eps,VioKKT);end;%(7)
 k=k+1; X0=X; Y0=Y; A(Omega)=X0(Omega)-Y0(Omega)*s;                 %(8)
 Dist=norm(A(Omega)-M(Omega),’fro’);                                %(9)
 if Dist <= delta   YT(Omega)=0;   else                            %(10)
 YT(Omega)=(M(Omega)+(A(Omega)-M(Omega))*delta/Dist-A(Omega))/s; %(11)
 end; EY=Y-YT; A=X0+(YT*2-Y0)/r; [U,D,V]=svd(A,0); D=D-eye(n)/r; %(12)
 D=max(D,0); XT=(U*D)*V’; EX=X-XT; DXM=XT(Omega)-M(Omega);         %(13)
 eps = norm(DXM,’fro’)/nM0; VioKKTX=max(max(abs(EX)))*r;           %(14)
 VioKKTY=max(max(abs(EY)));  VioKKT=max(VioKKTX,VioKKTY);          %(15)
 if eps <= tol    gamma=1;    end;                                 %(16)
 X=X0-EX*gamma; Y(Omega)=Y0(Omega)-EY(Omega)*gamma;                %(17)
end;                                                               %(18)
 fprintf(’It=%3d|X-M|/|M|=%9.2eVi0KKT=%9.2e\n’,k,eps/nM0,VioKKT);%(19)
 RelEr=norm((X-M),’fro’)/norm(M,’fro’);     toc;                   %(20)
 fprintf(’Relative error = %9.2e Rank(X)=%3d \n’,RelEr,rank(X)); %(21)
 fprintf(’Violation of KKT Condition = %9.2e \n’,VioKKT);          %(22)
```



## Relaxed PPA for Noise Matrix Completion

```
function PPA_RG(n,r,s,M,Omega,maxIt,gamma,delta,tol)               %(1)
X=zeros(n); Y=zeros(n); YT=zeros(n); A=zeros(n);                   %(2)
nM0=norm(M(Omega),’fro’); eps=100000; VioKKT=1; k=0; tic; sv=100;%(3)
%% Minimum nuclear norm solution by Relaxed PPA method             %(4)
while (eps > tol && k<= maxIt)                                     %(5)
if mod(k,10)==0                                                    %(6)
 fprintf(’It=%3d|X-M|/|M|=%9.2eVioKKT=%9.2e\n’,k,eps,VioKKT);end;%(7)
 k=k+1; X0=X; Y0=Y; A(Omega)=X0(Omega)-Y0(Omega)*s;                %(8)
 Dist=norm(A(Omega)-M(Omega),’fro’);                               %(9)
 if Dist <= delta   YT(Omega)=0;    else                          %(10)
 YT(Omega)=(M(Omega)+(A(Omega)-M(Omega))*delta/Dist-A(Omega))/s;%(11)
 end; EY=Y0-YT; A=X0+YT/r; [U,D,V]=svd(A,0); D=D-eye(n)/r;        %(12)
 D=max(D,0); XT=(U*D)*V’; EX=X0-XT; DXM=XT(Omega)-M(Omega);       %(13)
 eps=norm(DXM,’fro’)/nM0; VioKKT=max(max(abs(EX)))*r;             %(14)
 T1=EX(:)’*EX(:);     T2=EY(:)’*EY(:);                            %(15)
 T12=EY(:)’*EX(:);  EXOm=EX(Omega);    T3=EXOm(:)’*EXOm(:);       %(16)
 TA=T1*r+T2*s-T12; TB=T1*r+T2*s-T12*2+T3/s; alpha=TA*gamma/TB;   %(17)
 if eps <= tol   alpha=1;   end;  X=X0-EX*alpha;                  %(18)
 Y(Omega)=Y0(Omega)-(EY(Omega)-EXOm/s)*alpha;     end;            %(19)
fprintf(’It=%3d|X-M|/|M|=%9.2eVi0KKT=%9.2e\n’,k,eps/nM0,VioKKT);%(20)
 RelEr=norm((X-M),’fro’)/norm(M,’fro’);     toc;                  %(21)
 fprintf(’Relative error=%9.2e Rank(X)=%3d\n’,RelEr,rank(X));    %(22)
 fprintf(’Violation of KKT Condition=%9.2e\n’,VioKKT);            %(23)
```

带噪音矩阵完整化计算结果 Matlab-svd

| $1000 \times 1000$ | | Classical PPA | | Extended PPA | | Relaxed PPA | |
|---|---|---|---|---|---|---|---|
| $\sigma =$ | (rank, $\frac{m}{d_r}$, $\frac{m}{n^2}$) | No. It | CPU Sec. | No. It | CPU Sec. | No. It | CPU Sec. |
| 0.01 | (10, 6, 0.12) | 62 | 827.45 | 54 | 729.52 | 41 | 554.73 |
| | (50, 4, 0.39) | 32 | 427.17 | 27 | 362.30 | 21 | 288.03 |
| | (100, 3, 0.58) | 27 | 391.62 | 24 | 347.27 | 25 | 322.22 |
| 0.1 | (10, 6, 0.12) | 42 | 550.45 | 34 | 437.88 | 26 | 333.96 |
| | (50, 4, 0.39) | 20 | 255.15 | 17 | 225.86 | 13 | 166.26 |
| | (100, 3, 0.58) | 17 | 226.26 | 14 | 185.01 | 17 | 222.06 |

带噪音矩阵完整化计算结果 Propack-svd

| $1000 \times 1000$ | | Classical PPA | | Extended PPA | | Relaxed PPA | |
|---|---|---|---|---|---|---|---|
| $\sigma =$ | (rank, $\frac{m}{d_r}$, $\frac{m}{n^2}$) | No. It | CPU Sec. | No. It | CPU Sec. | No. It | CPU Sec. |
| 0.01 | (10, 6, 0.12) | 62 | 37.16 | 53 | 31.00 | 41 | 29.18 |
| | (50, 4, 0.39) | 31 | 56.54 | 27 | 41.57 | 20 | 41.80 |
| | (100, 3, 0.58) | 26 | 72.61 | 23 | 49.17 | 24 | 61.14 |
| 0.1 | (10, 6, 0.12) | 41 | 34.47 | 33 | 26.20 | 25 | 23.37 |
| | (50, 4, 0.39) | 19 | 42.59 | 16 | 30.30 | 13 | 29.01 |
| | (100, 3, 0.58) | 16 | 41.27 | 14 | 32.59 | 15 | 35.72 |



# References


[1] J. F. Cai, E. J. Candès and Z. W. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim., **20**, 1956-1982, 2010.

[2] B.S. He, A class of projection and contraction methods for monotone variational inequalities, Applied Mathematics and Optimization, **35**, 69-76, 1997.

[3] B.S. He, Inexact implicit methods for monotone general variational inequalities, Math. Program. Series A, **86**, 199-217, 1999.

[4] B. S. He, X. L. Fu and Z.K. Jiang, Proximal point algorithm using a linear proximal term, JOTA, **141**, 209-239, 2009.

[5] B.S He and L-Z Liao, *Improvements of some projection methods for monotone nonlinear variational inequalities*, JOTA, **112**, 111-128, 2002.

[6] B. S. He and M.-H. Xu, A general framework of contraction methods for monotone variational inequalities, Pacific J. Optimization, **4**, 195-212, 2008.

[7] B.S. He and X. M. Yuan, A contraction method with explicit proximal regularization for linearly constrained convex programming, Optimization Online, 2010.

[8] B. Qu and N.H. Xiu, A note on the CQ algorithm for the split feasiblity problem, Inverse Problems, **21** pp. 1655-1665, 2005.

[9] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Control Optim., **14**, 877-898, 1976.

[10] J.L. Zhao and Q.Z. Yang, Several solution methods for the split feasiblity problem, Inverse Problems, **21** pp. 1791-1799, 2005.

# 凸优化和单调变分不等式的收缩算法

## 第七讲: 基于增广 Lagrange 乘子法的 PPA 收缩算法

Augmented Lagrangian-based PPA contraction methods for constrained convex optimization

南京大学数学系　何炳生
hebma@nju.edu.cn



# 1 线性约束凸优化等价的单调变分不等式

考虑一般的线性等式约束凸优化问题

$$\min\{\theta(x) \mid Ax = b,\ x \in \mathcal{X}\}, \tag{1.1}$$

其中 $\theta(x)$ 是凸函数, $A \in \Re^{m\times n}$, $b \in \Re^m$, $\mathcal{X}$ 是 $\Re^n$ 中的闭凸集. 凸优化问题 (1.1) 的 Lagrange 函数是定义在 $\mathcal{X} \times \Re^m$ 上的

$$L(x,\lambda) = \theta(x) - \lambda^T(Ax-b).$$

记 $\Lambda = \Re^m$, 设 $(x^*,\lambda^*)$ 是 Lagrange 函数的一个鞍点, 便有

$$L_{\lambda\in\Lambda}(x^*,\lambda) \le L(x^*,\lambda^*) \le L_{x\in\mathcal{X}}(x,\lambda^*).$$

求 Lagrange 函数的一个鞍点等价于求 $(x^*,\lambda^*)$ 使其满足:

$$\left\{\begin{array}{lll} x^* \in \mathcal{X}, & \theta(x) - \theta(x^*) + (x-x^*)^T(-A^T\lambda^*) \ge 0, & \forall\, x \in \mathcal{X}, \\ \lambda^* \in \Lambda, & (\lambda-\lambda^*)^T(Ax^*-b) \ge 0, & \forall\, \lambda \in \Lambda, \end{array}\right. \tag{1.2}$$

通过定义

$$u = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F(u) = \begin{pmatrix} -A^T\lambda \\ Ax - b \end{pmatrix} \quad \text{和} \quad \Omega = \mathcal{X} \times \Lambda, \tag{1.3}$$

求解 (1.1) 相当于求解变分不等式

$$\text{VI}(\Omega, F) \quad u^* \in \Omega, \quad \theta(x) - \theta(x^*) + (u - u^*)^T F(u^*) \geq 0, \quad \forall\, u \in \Omega. \tag{1.4}$$

注意到, (1.3) 中的仿射算子 $F$ 是单调的.

对给定的常数 $s > 0$, 定义在 $\Omega = \mathcal{X} \times \Re^m$ 上的

$$\mathcal{L}_A(x, \lambda) = \theta(x) - \lambda^T(Ax - b) + \frac{1}{2s}\|Ax - b\|^2$$

是 (1.1) 的增广 Lagrange 函数. 增广Lagrange 乘子法 [1, 8, 11] 是求等式约束优化问题的很有效的方法之一, 对此, [10] 中第 17 章有详细论述.

求上述等式约束凸优化问题的经典增广 Lagrange 乘子法 (Augmented Lagrangian Method) 的 $k$-步迭代, 是从给定的 $\lambda^k$ 出发, 求

$$x^{k+1} = \text{Argmin}\{\mathcal{L}_A(x, \lambda^k) \,|\, x \in \mathcal{X}\},$$



然后以

$$\lambda^{k+1} = \lambda^k - \frac{1}{s}(Ax^{k+1} - b)$$

得到新的迭代点.

设 $\Omega^*$ 是 $\text{VI}(\Omega, F)$ 的解集合. 为考虑收缩算法需要, 我们记

$$\Lambda^* = \{\lambda^* \in \Lambda \,|\, (x^*, \lambda^*) \in \Omega^*\}.$$

如果用经典的增广 Lagrange 方法求解问题 (1.1), 对任意给定的 $b \in \Re^m$, 我们假设子问题

$$\boxed{\min\{\theta(x) + \tfrac{1}{2s}\|Ax - b\|^2 \,|\, x \in \mathcal{X}\}}$$

是容易求解的. 否则的话, 采用基于松弛增广 Lagrange 的收缩方法. 这时, 需要假设对任意给定的 $a \in \Re^n$, 问题

$$\boxed{\min\{\theta(x) + \tfrac{r}{2}\|x - a\|^2 \,|\, x \in \mathcal{X}\}}$$

的解是容易求得的.

## 2 收缩意义下的增广 Lagrange 乘子法

我们称定义在 $\Omega = \mathcal{X} \times \Re^m$ 上的

$$\mathcal{L}_A(x,\lambda) = \theta(x) - \lambda^T(Ax-b) + \frac{1}{2s}\|Ax-b\|^2$$

为等式约束的问题 $\min\{\theta(x) \mid Ax = b,\ x \in \mathcal{X}\}$ 的增广 Lagrange 函数. 以下是增广 Lagrange 乘子法的框架，它从给定的 $\lambda^k$ 开始.

**等式约束问题的增广 Lagrange 乘子法框架** 对给定的 $\lambda^k$, 先求

$$\tilde{x}^k = \text{Argmin}\{\theta(x) + \tfrac{1}{2s}\|(Ax-b) - s\lambda^k\|^2 \mid x \in \mathcal{X}\}, \tag{2.1a}$$

然后令

$$\tilde{\lambda}^k = \lambda^k - \tfrac{1}{s}(A\tilde{x}^k - b). \tag{2.1b}$$

经典的增广 Lagrange 乘子法以 $\lambda^{k+1} = \tilde{\lambda}^k$ 完成一次迭代.



我们用收缩算法的观点来分析增广 Lagrange 乘子法, 改成以

$$\lambda^{k+1} = \lambda^k - \gamma(\lambda^k - \tilde{\lambda}^k), \quad \gamma \in (0,2) \tag{2.2}$$

生成新的迭代点. 由 (2.1a) 生成的 $\tilde{x}^k \in \mathcal{X}$ 满足

$$\theta(x) - \theta(\tilde{x}^k) + (x-\tilde{x}^k)^T\{-A^T\lambda^k + \tfrac{1}{s}A^T(A\tilde{x}^k - b)\} \ge 0, \ \ \forall x \in \mathcal{X}.$$

将 (2.1b) 中的 $\tilde{\lambda}^k = \lambda^k - \frac{1}{s}(A\tilde{x}^k - b)$ 代入上式就有

$$\tilde{x}^k \in \mathcal{X}, \quad \theta(x) - \theta(\tilde{x}^k) + (x-\tilde{x}^k)^T\big(-A^T\tilde{\lambda}^k\big) \ge 0, \ \ \forall x \in \mathcal{X}. \tag{2.3}$$

把 (2.3) 和 (2.1b) 组合在一起, 就是 $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k) \in \Omega$, 并且

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} + \begin{pmatrix} 0 \\ s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \ge 0, \ \forall u \in \Omega. \tag{2.4}$$

将任意的 $u^* = (x^*, \lambda^*)$ 代入上式中的 $u \in \Omega$, 并利用 $F(u)$ 的表达式就有

$$(\tilde{\lambda}^k - \lambda^*)^T(\lambda^k - \tilde{\lambda}^k) \ge \frac{1}{s}\{(\tilde{u}^k - u^*)^T F(\tilde{u}^k) + \theta(\tilde{x}^k) - \theta(x^*)\}. \tag{2.5}$$

利用 $F$ 的单调性和

$$\theta(\tilde{x}^k) - \theta(x^*) + (\tilde{u}^k - u^*)^T F(u^*) \geq 0,$$

推得 (2.5) 式的右端非负. 所以有

$$(\lambda^k - \lambda^*)^T(\lambda^k - \tilde{\lambda}^k) \geq \|\lambda^k - \tilde{\lambda}^k\|^2, \quad \forall\, \lambda^* \in \Lambda^*. \tag{2.6}$$

用 (2.2) 生成新的迭代点 $\lambda^{k+1}$, 在收缩意义下我们通常取 $\gamma \in [1,2)$, 根据 (2.6) 得到

$$\begin{aligned}
\|\lambda^{k+1} - \lambda^*\|^2 &= \|(\lambda^k - \lambda^*) - \gamma(\lambda^k - \tilde{\lambda}^k)\|^2 \\
&= \|\lambda^k - \lambda^*\|^2 - 2\gamma(\lambda^k - \lambda^*)^T(\lambda^k - \tilde{\lambda}^k) + \gamma^2\|\lambda^k - \tilde{\lambda}^k\|^2 \\
&\leq \|\lambda^k - \lambda^*\|^2 - \gamma(2-\gamma)\|\lambda^k - \tilde{\lambda}^k\|^2.
\end{aligned}$$

The sequence $\{\lambda^k\}$ (dual variable) generated by Augmented Lagrangian Method is Fejér monotone.

上述性质说明, 增广 Lagrange 乘子法是关于对偶变量 $\lambda$ 的 PPA 算法. 用收缩算法的观点来考虑问题, 迭代式 (2.2) 中的 $\gamma$ 可以在区间 $(0,2)$ 中自由选取. 在实际计算中, 我们建议取 $\gamma \in [1.2, 1.8]$.



# 3 基于增广 Lagrange 乘子法的 PPA 算法

在实际问题中, 精确求解子问题 (2.1a) 往往是花费很大的. 我们还是假设只有形似 $\min\{\theta(x) + \frac{r}{2}\|x-a\|^2 \,|\, x \in \mathcal{X}\}$ 的问题是容易求解的. 如果对 (2.1a) 中的目标函数加上 Proximal 项 $\dfrac{r}{2}\|x - x^k\|^2$, 要处理的子问题就成为

$$\min\,\{\theta(x) + \frac{1}{2s}\|(Ax-b) - s\lambda^k\|^2 + \frac{r}{2}\|x - x^k\|^2 \,|\, x \in \mathcal{X}\}. \tag{3.1}$$

再将 (3.1) 中的 $\dfrac{1}{2s}\|(Ax-b) - s\lambda^k\|^2$ 在 $x^k$ 处做线性化近似就是

$$\frac{1}{2s}\|(Ax^k - b) - s\lambda^k\|^2 + \Big(\frac{1}{s}A^T[(Ax^k-b) - s\lambda^k]\Big)^T(x - x^k).$$

对 (3.1) 中的二次项线性化以后, 子问题就变成

$$\min\,\{\theta(x) + \Big(\frac{1}{s}A^T[(Ax^k-b) - s\lambda^k]\Big)^T x + \frac{r}{2}\|x - x^k\|^2 \,|\, x \in \mathcal{X}\}. \tag{3.2}$$

基于增广 Lagrange 乘子法的 PPA 算法, 只对 (3.1) 中的二次函数做线性化处理. 每步迭代从给定的 $u^k = (x^k, \lambda^k)$ 开始, 生成 $\tilde{u}^k \in \Omega$.

**对二次函数线性化处理** 对给定的 $(x^k,\lambda^k)$, 先求

$$\tilde{x}^k = \text{Argmin}\{\theta(x)+\big(\frac{1}{s}A^T[(Ax^k-b)-s\lambda^k]\big)^T x+\frac{r}{2}\|x-x^k\|^2 \,|\, x\in\mathcal{X}\} \quad \text{(3.3a)}$$

然后令

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}(A\tilde{x}^k - b). \quad \text{(3.3b)}$$

我们按统一框架考察生成的预测点 $\tilde{u}^k$. 由 (3.3a) 生成的 $\tilde{x}^k\in\mathcal{X}$ 满足

$$\theta(x)-\theta(\tilde{x}^k)+(x-\tilde{x}^k)^T\{-A^T\lambda^k+\frac{1}{s}A^T(Ax^k-b)+r(\tilde{x}^k-x^k)\}\ge 0,\ \forall x\in\mathcal{X}.$$

将 (3.3b) 中的 $\tilde{\lambda}^k=\lambda^k-\frac{1}{s}(A\tilde{x}^k-b)$ 代入上式就有 $\tilde{x}^k\in\mathcal{X}$, 并对$x\in\mathcal{X}$, 有

$$\theta(x)-\theta(\tilde{x}^k)+(x-\tilde{x}^k)^T\big(-A^T\tilde{\lambda}^k+(rI_n-\frac{1}{s}A^TA)(\tilde{x}^k-x^k)\big)\ge 0, \quad \text{(3.4)}$$

把 (3.4) 和 (3.3b) 组合在一起, 就是 $\tilde{u}^k=(\tilde{x}^k,\tilde{\lambda}^k)\in\Omega$, 并对所有的 $(x,\lambda)\in\Omega$,



均有

$$\theta(x)-\theta(\tilde{x}^k)+\begin{pmatrix} x-\tilde{x}^k \\ \lambda-\tilde{\lambda}^k\end{pmatrix}^T\left\{\begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k-b\end{pmatrix}+\begin{pmatrix} (rI_n-\frac{1}{s}A^TA)(\tilde{x}^k-x^k) \\ s(\tilde{\lambda}^k-\lambda^k)\end{pmatrix}\right\}\ge 0. \quad \text{(3.5)}$$

我们记

$$d(u^k,\tilde{u}^k)=\begin{pmatrix} (rI_n-\frac{1}{s}A^TA)(x^k-\tilde{x}^k) \\ s(\lambda^k-\tilde{\lambda}^k)\end{pmatrix}. \quad \text{(3.6)}$$

并利用 $F(u)$ 的表达式, 就有

$$\theta(x)-\theta(\tilde{x}^k)+(u-\tilde{u}^k)^T(F(\tilde{u}^k)-d(u^k,\tilde{u}^k))\ge 0,\ \ \forall\, u\in\Omega. \quad \text{(3.7)}$$

将任意的 $u^*=(x^*,\lambda^*)$ 代入上式中的 $(x,\lambda)\in\Omega$, 并

$$(\tilde{u}^k-u^*)^T d(u^k,\tilde{u}^k)\ge \theta(\tilde{x}^k)-\theta(x^*)+(\tilde{u}^k-u^*)^T F(\tilde{u}^k). \quad \text{(3.8)}$$

利用 $F$ 的单调性和 $\theta(\tilde{x}^k)-\theta(x^*)+(\tilde{u}^k-u^*)^TF(u^*)\ge 0$, 上式右端非负. 进而得到

$$(u^k-u^*)^T d(u^k,\tilde{u}^k)\ge (u^k-\tilde{u}^k)^T d(u^k,\tilde{u}^k). \quad \text{(3.9)}$$

**对参数 $r, s$ 的要求** 对给定的 $s > 0$, 选取 $r$ 使

$$rs > \|A^T A\|. \tag{3.10}$$

在上述条件下, 矩阵

$$G = \begin{pmatrix} rI_n - \frac{1}{s}A^T A & 0 \\ 0 & sI_m \end{pmatrix}, \tag{3.11}$$

是正定的. 利用 (3.6), 就有

$$d(u^k, \tilde{u}^k) = G(u^k - \tilde{u}^k). \tag{3.12}$$

不等式 (3.9) 就化成

$$(u^k - u^*)^T G(u^k - \tilde{u}^k) \geq \|u^k - \tilde{u}^k\|_G^2. \tag{3.13}$$

直接取 $u^{k+1} = \tilde{u}^k$ 为新的迭代点, 就得到 $G$-模下的 PPA 算法.

**基于增广 Lagrange 乘子法的 PPA 算法**



PPA 算法用迭代式

$$u^{k+1} = u^k - \gamma(u^k - \tilde{u}^k), \quad \gamma \in (0, 2) \tag{3.14}$$

产生新的迭代点 $u^{k+1}$, 我们一般取 $\gamma \in [1, 2)$. 利用 (3.13), 序列 $\{u^k\}$ 满足

$$\|u^{k+1} - u^*\|_G^2 \leq \|u^k - u^*\|_G^2 - \gamma(2-\gamma)\|u^k - \tilde{u}^k\|_G^2.$$

这是保证基于增广 Lagrange 乘子法的 PPA 算法收敛的关键不等式.

使用这一节的算法, 参数 $r, s$ 需要满足条件 $rs > \|A^T A\|$ (见 (3.10)). 因此, 适合用来处理 $\|A^T A\|$ 容易估算的问题.

## 4 基于增广 Lagrange 乘子法的收缩算法

这一节的基于增广 Lagrange 乘子法的收缩算法, 同样是只对 (3.1) 中的二次函数做线性化处理. 每步迭代从给定的 $u^k = (x^k, \lambda^k)$ 开始, 还是以 (3.3) 生成 $\tilde{u}^k \in \Omega$. 由上一节的分析, 我们得到

$$(u^k - u^*)^T d(u^k, \tilde{u}^k) \geq (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k), \tag{4.1}$$

其中

$$d(u^k,\tilde{u}^k)=\begin{pmatrix}(rI_n-\frac{1}{s}A^TA)(x^k-\tilde{x}^k)\\ s(\lambda^k-\tilde{\lambda}^k)\end{pmatrix} \tag{4.2}$$

我们记

$$H=\begin{pmatrix}rI_n & 0\\ 0 & sI_m\end{pmatrix}. \tag{4.3}$$

并考虑用

$$M(u^k-\tilde{u}^k)=H^{-1}d(u^k,\tilde{u}^k) \tag{4.4}$$

作为寻查方向. 这时

$$M=\begin{pmatrix}I_n-\frac{1}{rs}A^TA & 0\\ 0 & I_m\end{pmatrix}. \tag{4.5}$$

我们并不要求 $M$ 正定, 因此不再要求 $rs>\|A^TA\|$, 也不能直接将 $\tilde{u}^k$ 取作新的迭代点. 换句话说, 基于增广 Lagrange 乘子法的收缩算法是一种预测一校正方法, 以 (3.3) 生成的 $\tilde{u}^k\in\Omega$ 只是一个预测点.



**生成预测点时对参数 $r,s$ 的要求** 对给定的 $s>0$, 要求 $r$ 满足

$$\|A^TA(x^k-\tilde{x}^k)\|\le rs\nu\|x^k-\tilde{x}^k\|,\quad \nu\in(0,1). \tag{4.6}$$

如果条件 $rs>\|A^TA\|$ (见 (3.10)) 满足, 则条件 (4.6) 自然满足. 然而, 这里只要求在计算的每次迭代中验证条件 (4.6) 是否满足.

我们考虑 $H$ 模下的收缩算法. 注意到不等式 (4.1) 可以写成

$$(u^k-u^*)^THM(u^k-\tilde{u}^k)\ge(u^k-\tilde{u}^k)^THM(u^k-\tilde{u}^k),\ \forall\, u^*\in\Omega^*. \tag{4.7}$$

**Lemma 4.1** *对给定的 $u^k=(x^k,\lambda^k)$, 设 $\tilde{u}^k\in\Omega$ 由 (3.3) 生成. 在条件 (4.6) 满足的情况下, 我们有*

$$\begin{aligned}&(u^k-\tilde{u}^k)^THM(u^k-\tilde{u}^k)\\ &\quad\ge\ \frac{1}{2}\{\|M(u^k-\tilde{u}^k)\|_H^2+(1-\nu^2)\|u^k-\tilde{u}^k\|_H^2\}.\end{aligned} \tag{4.8}$$

**证明.** 我们考察

$$2(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)-\|d(u^k,\tilde{u}^k)\|_H^2.$$

由于

$$\begin{aligned} & 2(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)-\|M(u^k-\tilde{u}^k)\|_H^2 \\ = \;& \big(2(u^k-\tilde{u}^k)-M(u^k-\tilde{u}^k)\big)^T HM(u^k-\tilde{u}^k). \end{aligned} \tag{4.9}$$

利用 $H$ 和 $d(u^k,\tilde{u}^k)$ 的定义(见 (4.3) 和 (4.4)), 对上式的右端进行处理, 由于

$$M(u^k-\tilde{u}^k) \;=\; \begin{pmatrix} (I_n-\frac{1}{rs}A^TA)(x^k-\tilde{x}^k) \\ (\lambda^k-\tilde{\lambda}^k) \end{pmatrix}$$

和

$$2(u^k-\tilde{u}^k)-M(u^k-\tilde{u}^k) \;=\; \begin{pmatrix} (I_n+\frac{1}{rs}A^TA)(x^k-\tilde{x}^k) \\ (\lambda^k-\tilde{\lambda}^k) \end{pmatrix}.$$



将这些代入 (4.9) 的右端就得到

$$\begin{aligned} & 2(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)-\|M(u^k-\tilde{u}^k)\|_H^2 \\ = \;& r\big(\|x^k-\tilde{x}^k\|^2-\frac{1}{r^2s^2}\|A^TA(x^k-\tilde{x}^k)\|^2\big)+s\|\lambda^k-\tilde{\lambda}^k\|^2. \end{aligned}$$

在条件 (4.6) 满足的情况下, 从上式得到

$$\begin{aligned} & 2(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k)-\|M(u^k-\tilde{u}^k)\|_H^2 \\ \geq \;& (1-\nu^2)r\|x^k-\tilde{x}^k\|^2+s\|\lambda^k-\tilde{\lambda}^k\|^2. \end{aligned} \tag{4.10}$$

从上式马上得到 (4.8), 引理证明完毕.

我们考虑 $H$ 模下的收缩算法. 先考虑取单位步长的**初等的收缩算法**.

**初等的收缩算法** 对给定的 $u^k$ 和用 (3.3) 生成的 $\tilde{u}^k$, 用

$$u^{k+1}=u^k-M(u^k-\tilde{u}^k) \tag{4.11}$$

生成新的迭代点. 由矩阵 $M$ 的结构 (4.5), 采用初等收缩算法时 $\lambda^{k+1}=\tilde{\lambda}^k$.

根据迭代公式 (4.11) 并使用 (4.1), 就有

$$\begin{aligned} & \|u^k - u^*\|_H^2 - \|u^{k+1} - u^*\|_H^2 \\ = \;& \|u^k - u^*\|_H^2 - \|(u^k - u^*) - M(u^k - \tilde{u}^k)\|_H^2 \\ = \;& 2(u^k - u^*)^T HM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_H^2 \\ \geq \;& 2(u^k - \tilde{u}^k)^T HM(u^k - \tilde{u}^k) - \|M(u^k - \tilde{u}^k)\|_H^2. \end{aligned} \tag{4.12}$$

以引理 4.1 的结论 (4.8) 代入 (4.12), 就有下面的定理:

**Theorem 4.1** *Let the condition* (4.6) *be satisfied. Then the sequence* $\{u^k = (x^k, \lambda^k)\}$ *generated by the elementary contraction method satisfies*

$$\|u^{k+1} - u^*\|_H^2 \leq \|u^k - u^*\|_H^2 - (1 - \nu^2)\|u^k - \tilde{u}^k\|_H^2. \tag{4.13}$$

定理 4.1 中的不等式 (4.13) 是保证初等收缩算法收敛的关键不等式.

再考虑通过计算步长确定下一个迭代点的**一般的收缩算法**.

**一般的收缩算法** 对给定的 $u^k$ 和 (3.3) 生成的 $\tilde{u}^k$, 用

$$u(\alpha) = u^k - \alpha M(u^k - \tilde{u}^k) \tag{4.14}$$



产生依赖于步长 $\alpha$ 的迭代点. 同样, 对任意给定的 $u^* \in \Omega^*$, 我们定义

$$\vartheta(\alpha) := \|u^k - u^*\|_H^2 - \|u(\alpha) - u^*\|_H^2 \tag{4.15}$$

和

$$q(\alpha) = 2\alpha(u^k - \tilde{u}^k)^T HM(u^k - \tilde{u}^k) - \alpha^2\|M(u^k - \tilde{u}^k)\|_H^2. \tag{4.16}$$

利用 (4.14) 和 (4.15) 中 $\vartheta(\alpha)$ 的定义以及 (4.1), 可以证明

$$\vartheta(\alpha) \geq q(\alpha) \tag{4.17}$$

同样, 注意到 (4.16) 中的 $q(\alpha)$ 是 $\alpha$ 的二次函数, 它在

$$\alpha^* = \frac{(u^k - \tilde{u}^k)^T HM(u^k - \tilde{u}^k)}{\|M(u^k - \tilde{u}^k)\|_H^2} \tag{4.18}$$

时取得极大值. 从 (4.8) 式知, 在条件 (4.6) 满足的情况下, $\alpha_k^* \geq 1/2$. 在实际计算中, 我们取

$$u^{k+1} = u^k - \gamma\alpha_k^* M(u^k - \tilde{u}^k), \tag{4.19}$$

为新的迭代点, 其中 $\gamma \in [1,2)$ 称为松弛因子. 由 (4.15) 和 (4.17), 有

$$\begin{aligned}\|u^{k+1}-u^*\|_H^2 &\le \|u^k-u^*\|_H^2 - q(\gamma\alpha_k^*)\\ &\le \|u^k-u^*\|_H^2 - \gamma(2-\gamma)\alpha_k^*(u^k-\tilde{u}^k)^T HM(u^k-\tilde{u}^k).\end{aligned} \quad (4.20)$$

根据上式, 由 (4.8) 和 $\alpha_k^* \ge 1/2$, 得到

**Theorem 4.2** *The sequence $\{u^k=(x^k,\lambda^k)\}$ generated by the general contraction method satisfies*

$$\|u^{k+1}-u^*\|_H^2 \le \|u^k-u^*\|_H^2 - \frac{\gamma(2-\gamma)(1-\nu^2)}{4}\|u^k-\tilde{u}^k\|_H^2. \quad (4.21)$$

定理 4.2 中的不等式 (4.21) 是保证一般收缩算法收敛的关键不等式.

此外, 由 (4.18), (4.20) 和 $\alpha_k^* > 1/2$, 会有

$$\|u^{k+1}-u^*\|_H^2 \le \|u^k-u^*\|_H^2 - \frac{\gamma(2-\gamma)}{4}\|M(u^k-\tilde{u}^k)\|_H^2.$$



# 5 信息技术领域优化问题中的应用

我们用这一讲 §3 的方法求解第四讲中提到的算例.

## 5.1 相关性矩阵 (Correlation Matrix) 校正中的应用

对给定的对称矩阵 $C$, 求 $F$-模下与 $C$ 距离最近的相关性矩阵, 其数学表达式是

$$\min\Big\{\frac{1}{2}\|X-C\|_F^2 \mid \mathrm{diag}(X)=e,\ X\in S_+^n\Big\}, \quad (5.1)$$

其中 $e$ 表示每个分量都为 1 的 $n$-维向量, $S_+^n$ 表示 $n\times n$ 正半定锥的集合. 问题 (5.1) 是形如 (1.1) 的等式约束凸优化问题, 其中 $\|A^TA\|=1$.

我们用 $z\in\Re^n$ 作为等式约束 $\mathrm{diag}(X)=e$ 的 Lagrange 乘子.

**PPA 算法求解问题(5.1)**

对给定的 $(X^k,z^k)$, 用 (3.3) 产生$(\tilde{X}^k,\tilde{z}^k)$:

1. Finding $\tilde{X}^k$ which is the solution of the following minimization problem

$$\min\{\frac{1}{2}\|X - C\|_F^2 + \frac{r}{2}\|X - [X^k + \frac{1}{r}\,\mathrm{diag}(z^{k+\frac{1}{2}})]\|_F^2 | X \in S_+^n\}, \tag{5.2}$$

where

$$z^{k+\frac{1}{2}} = z^k - \frac{1}{s}(\mathrm{diag}(X^k) - e).$$

2. Setting

$$\tilde{z}^k = z^k - \frac{1}{s}(\mathrm{diag}(\tilde{X}^k) - e). \tag{5.3}$$

子问题 (5.2) 求解的具体做法: 化为等价问题

$$\min\{\frac{1}{2}\|X - \frac{1}{1+r}[rX^k + \mathrm{diag}(z^{k+\frac{1}{2}}) + C]\|_F^2 | X \in S_+^n\}.$$

因此我们只要考虑如何求解

$$\tilde{X}^k = \text{Argmin}\{\frac{1}{2}\|X - A\|_F^2 \mid X \in S_+^n\}. \tag{5.4}$$

实际上, 将对称矩阵 $A$ 做标准特征值– 特征向量分解

$$A = V\Lambda V^T, \qquad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n). \tag{5.5}$$



通过

$$\tilde{X} = V\tilde{\Lambda}V^T, \qquad \tilde{\Lambda} = \mathrm{diag}(\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_n),$$

就得到 $\tilde{X}$, 其中

$$\tilde{\lambda}_j = \max\{0, \lambda_j\}.$$

因此, 每次迭代的主要工作量是做 (5.5) 中的特征值 (特征向量) 分解 .

**数值试验** 为生成试验例子, 只要给定对称矩阵 $C$.

C=rand(n,n);  C=(C'+C)-ones(n,n) + eye(n)

这样的矩阵 $C$ 的对角元在 $(0,2)$ 之间, 非对角元在 $(-1,1)$ 之间。

Code 7.a. Matlab code for Creating the test examples

```
clear;  close all;
n = 1000;      tol=1e-5;      r=2.0;    s=1.05/r;     gamma=1.5;
rand('state',0);  C=rand(n,n);   C=(C'+C)-ones(n,n) + eye(n);
```

Code 7.1. Matlab code of ALM based classical PPA

```
%%%     Classical PPA for calibrating correlation matrix          %(1)
function PPAC(n,C,r,s,tol)                                        %(2)
X=eye(n);      y=zeros(n,1);    tic;      %%  The initial iterate %(3)
stopc=1;                             k=0;                         %(4)
while (stopc>tol && k<=100)         %% Beginning of an Iteration  %(5)
 if mod(k,20)==0 fprintf(’ k=%4d   epsm=%9.3e  \n’,k,stopc); end; %(6)
    X0=X;     y0=y;     k=k+1;                                    %(7)
    ya=y0 - (diag(X0)-ones(n,1))/s;                               %(8)
    A=(X0*r + C + diag(ya))/(1+r);                                %(9)
    [V,D]=eig(A);   D=max(0,D);  XT=(V*D)*V’;     EX=X0-XT;      %(10)
    yt=y0 - (diag(XT)-ones(n,1))/s;               EY=y0-yt       %(11)
    ex=max(max(abs(EX)));  ey=max(abs(EY));   stopc=max(ex,ey);  %(12)
    X= XT;                  y=yt;                                %(13)
end;                                        % End of an Iteration %(14)
toc;                        TB = max(abs(diag(X-eye(n))));       %(15)
fprintf(’ k=%4d  epsm=%9.3e   max|X_jj - 1|=%8.5f \n’,k,stopc,TB); %%
```

做 (5.5) 中的特征值 (特征向量) 分解, 在上述程序中的第 $(10)$ 行用 Matlab 中的语句 [V,D]=eig(A) 实现的, 这是一个计算量大概 $9n^3$ 的运算.

将 Classical PPA 改成 Extended PPA, 只要将第 $(13)$ 行改一下.



Code 7.2 Matlab code of ALM based extended PPA

```
%%%     Extended PPA for calibrating correlation matrix           %(1)
function PPAE(n,C,r,s,tol,gamma)                                  %(2)
X=eye(n);      y=zeros(n,1);    tic;      %%  The initial iterate %(3)
stopc=1;                             k=0;                         %(4)
while (stopc>tol && k<=100)         %% Beginning of an Iteration  %(5)
 if mod(k,20)==0 fprintf(’ k=%4d   epsm=%9.3e  \n’,k,stopc); end; %(6)
    X0=X;     y0=y;     k=k+1;                                    %(7)
    ya=y0 - (diag(X0)-ones(n,1))/s;                               %(8)
    A=(X0*r + C + diag(ya))/(1+r);                                %(9)
    [V,D]=eig(A);   D=max(0,D);  XT=(V*D)*V’;     EX=X0-XT;      %(10)
    yt=y0 - (diag(XT)-ones(n,1))/s;               EY=y0-yt       %(11)
    ex=max(max(abs(EX)));  ey=max(abs(EY));   stopc=max(ex,ey);  %(12)
    X=X0 - EX*gamma;          y=y0- EY*gamma;                    %(13)
end;                                        % End of an Iteration %(14)
toc;                        TB = max(abs(diag(X-eye(n))));       %(15)
fprintf(’ k=%4d  epsm=%9.3e   max|X_jj - 1|=%8.5f \n’,k,stopc,TB); %%
```

两个不同的方法, 程序都很简单, 用不了几行. 两个程序不同的地方仅仅是第 (13) 行有些差别, 取 $\gamma = 1.5$ 的方法效果却有明显的提高。

矩阵校正问题 (5.1)–使用 Matlab 中的 eig 子程序

| $n \times n$ Matrix | Classical PPA | | Extended PPA | |
|---|---|---|---|---|
| $n =$ | No. It | CPU Sec. | No. It | CPU Sec. |
| 100 | 29 | 0.34 | 21 | 0.25 |
| 200 | 32 | 2.26 | 24 | 1.68 |
| 500 | 38 | 20.70 | 26 | 14.04 |
| 800 | 41 | 86.75 | 29 | 61.28 |
| 1000 | 47 | 182.82 | 30 | 117.08 |
| 2000 | 65 | 1696.50 | 41 | 1076.04 |

The extended PPA converges faster than the classical PPA.

$$\frac{\text{It. No. of Extended PPA}}{\text{It. No. of Classical PPA}} \approx 65\%.$$



♣ 关于相关系数矩阵校正的程序在附件的 Codes-07 的文件夹“*矩阵校正*”中. 只要运行 demo.m, 输入 n 就可以了. 其中的 ALM_PPAC.m 和 ALM_PPAE.m 分别是 Classical PPA 和 Extended PPA 的子程序.

确实, 用这一讲介绍的 PPA 方法求解相关矩阵矩阵校正问题, 每步迭代的主要计算工作量是对一个对称矩阵用 Matlab 中的标准子程序做 [V,D]=eig(A). 如果改用 Kim TOH 写的 mexeig 做 [V,D]=mexeig(A), 计算时间大为节省.

矩阵校正问题 (5.1)–特征值分解使用 mexeig

| $n \times n$ Matrix | Classical PPA | | Extended PPA | |
|---|---|---|---|---|
| $n =$ | No. It | CPU Sec. | No. It | CPU Sec. |
| 100 | 29 | 0.14 | 21 | 0.10 |
| 200 | 32 | 0.57 | 24 | 0.42 |
| 500 | 38 | 5.64 | 26 | 3.92 |
| 800 | 41 | 20.05 | 29 | 14.31 |
| 1000 | 47 | 41.82 | 30 | 27.10 |
| 2000 | 65 | 401.94 | 41 | 255.37 |

## 5.2 矩阵完整化方面的应用

设 $M$ 是一个 $m \times n$ 矩阵, $\Pi$ 是矩阵元素的指标集.

$$\Pi = \{(ij) \,|\, i \in \{1, 2, \ldots, m\},\ j \in \{1, 2, \ldots, n\}\}.$$

矩阵完整化问题是由部分信息获取全部信息. 在适当(实际问题具备的)条件下, 大多数不完整信息的低秩矩阵可以通过求解松弛问题

$$\min\{\|X\|_* \mid X_{ij} = M_{ij},\ (ij) \in \Pi\} \tag{5.6}$$

得到精确恢复. 其中 $\|X\|_*$ 表示矩阵 $X$ 的奇异值的和. 通常称为矩阵 $X$ 的核模—Nuclear Norm.

问题 (5.6) 是形如 (1.1) 的等式约束凸优化问题, 其中 $\|A^T A\| = 1$. 我们将 (5.6) 的等式约束记为 $X_\Pi = M_\Pi$，并用 $Z \in \Re^{m\times n}$ 作为相应的 Lagrange 乘子.

**PPA 算法求解问题 (5.6)**

对给定的 $(X^k, Z^k)$, 用 (3.3) 产生$(\tilde{X}^k, \tilde{Z}^k)$:



1. Finding $\tilde{X}^k$ which is the solution of the following linear variational inequality

$$\min\ \{\|X\|_* + \frac{r}{2}\|X - [X^k + \frac{1}{r}Z_\Pi^{k+\frac{1}{2}}]\|_F^2\} \tag{5.7}$$

   where

$$Z_\Pi^{k+\frac{1}{2}} = Z_\Pi^k - \frac{1}{s}(X_\Pi^k - M_\Pi).$$

2. Updating $\tilde{Z}^k$ by

$$\tilde{Z}_\Pi^k = Z_\Pi^k - \frac{1}{s}(\tilde{X}_\Pi^k - M_\Pi).$$

**子问题 (5.7) 求解的具体做法:** 只需考虑如何求解

$$\tilde{X}^k = \text{Argmin}\{\frac{1}{r}\|X\|_* + \frac{1}{2}\|X - A\|_F^2\}. \tag{5.8}$$

我们将 $A$ 做 SVD 分解

$$A = U\Lambda V^T,$$

并记

$$\tilde{X} = U\tilde{\Lambda}V^T. \tag{5.9}$$

注意到

$$\frac{1}{r}\|\tilde{X}\|_* + \frac{1}{2}\|\tilde{X} - A\|_F^2 = \frac{1}{r}\|\tilde{\Lambda}\|_* + \frac{1}{2}\|\tilde{\Lambda} - \Lambda\|_F^2.$$

通过

$$\tilde{\lambda}_j = \lambda_j - \min(\lambda_j, \frac{1}{r}), \tag{5.10}$$

就能得到对角矩阵 $\tilde{\Lambda}$ 的对角元 $\tilde{\lambda}_j$, 代入 (5.9) 就得到 (5.8) 的解 $\tilde{X}^k$. 同样, 每次迭代的主要工作量是做一个矩阵的 SVD 分解.

如果以 $\lambda$ 和 $\tilde{\lambda}$ 分别表示对角矩阵 $\Lambda$ 和 $\tilde{\Lambda}$ 的对角元生成的向量, 由于 $\lambda$ 是非负向量, 关系式 (5.10) 也可以写成 Shrinkage 的形式

$$\tilde{\lambda} = \lambda - P_{B_\infty^{1/r}}[\lambda],$$

其中 $B_\infty^{1/r}$ 是无穷模下半径为 $1/r$ 的“圆”（一个立方体）.

**结论**: 将线性约束的凸优化问题转换成单调变分不等式, 再用这一讲 §3 中介绍的基于增广 Lagrange 乘子法的 PPA 算法求解, 每次迭代中要求解的子问题, 数值代数中都有确定的成熟的方法求解 !



**数值试验**　　数值试验例子取自[2]

一个秩为 $ra$ 的 $n \times n$ 的自由度是 $d_{ra} := ra(2n - ra)$.

**生成试验问题:**

- 先用高斯同分布 (Gaussian i.i.d) 独立生成两个 $n \times ra$ 的矩阵 $M_1$ 和 $M_2$, 然后令 $M = M_1 M_2^T$, 则 $n \times n$ 矩阵 $M$ 的秩为 $ra$.
- 随机选定 $M$ 的 $m$ 个元素作为已知元素, 这些元素的下标集为 $\Pi$.

**计算结果:**

- 矩阵完整化问题的难度与比率 $m/d_{ra}$ 和 $m/n^2$ 都有关系.
- 分别用 Classical PPA 和 Extended PPA ($\gamma = 1.5$) 进行计算.
- 正定矩阵 $G$ (see (3.11)) 中的参数 $r$, $s$ 分别取 $rs = 1.01$ 和 $r = 0.005$.
- 停机准则采用相对误差 $\|X_\Pi^k - M_\Pi\|_F / \|M_\Pi\|_F \le 10^{-4}$.

我们用 $\gamma = 1.5$ 的Extended PPA 求解, 由于 KKT 条件 Primal 部分((3.5) 的上半

部分) 的不满足量是

$$(rI_n - \frac{1}{s}A^TA)(\tilde{x}^k - x^k).$$

对这个具体问题, $\|A^TA\| = 1$ 并且 $rs \approx 1$, 所以我们检查

$$\text{KKT-Violation} := \max\{r \max_{ij} |X_{ij}^k - \tilde{X}_{ij}^k|\}$$

并在计算结果中列出.

计算时间依赖于用什么 SVD 子程序. 我们分别列出用 Matlab 中的标准 SVD 以及 PROPACK-SVD 的不同计算效果.

♣ 矩阵完整化的程序在附件的 Codes-07 的文件夹"*矩阵完整化*"中. 只要运行 demo.m 就可以了. 要对不同情形试验, 只要在 demo.m 中用 % 做适当选择. 其中的 PPAC.m 和 PPAE.m 分别是 Classical PPA 和 Extended PPA 的子程序.



Code 7.b. Creating the test examples of Matrix Completion

```
%%  Creating the test examples of the matrix Completion problem     %(1)
clear all;  clc                                                      %(2)
maxIt=100;         tol = 1e-4;                                       %(3)
r=0.005;           s=1.01/r;         gamma=1.5;                      %(4)
  n=200;             ra = 10;          oversampling = 5;             %(5)
% n=1000;    ra=100;     oversampling = 3; %%  Iteration No. 31      %(6)
% n=1000;    ra=50;      oversampling = 4; %%  Iteration No. 36      %(7)
% n=1000;    ra=10;      oversampling = 6; %%  Iteration No. 78      %(8)
%% Generating the test problem                                       %(9)
rs = randseed;         randn('state',rs);                           %(10)
M=randn(n,ra)*randn(ra,n);            %% The matrix will be completed %(11)
df =ra*(n*2-ra);                      %% The freedom of the matrix   %(12)
mo=oversampling;                                                    %(13)
m =min(mo*df,round(.99*n*n));           %% No. of the known elements %(14)
Omega= randsample(n^2,m);               %% Define the subset Omega   %(15)
fprintf('Matrix: n=%4d  Rank(M)=%3d  Oversampling=%2d \n',n,ra,mo);%(16)
```

我们只对ALM-based extended PPA 给出程序, 若用ALM-based classical PPA, 只要把下面的第(16) 行改成 `X = XT` 第(17) 行改成 `Y(Omega) = YT(Omega)`

Code 7.3. ALM-based extended PPA for Matrix Completion Problem

```
function PPAE(n,r,s,M,Omega,maxIt,tol,gamma)    %  Itial Process %%(1)
X=zeros(n);      Y=zeros(n);      YT=zeros(n);                      %(2)
  nM0=norm(M(Omega),'fro');    eps=1;  VioKKT=1;  k=0;   tic;       %(3)
%% Minimum nuclear norm solution by PPA method                      %(4)
while  (eps > tol && k<= maxIt)                                     %(5)
 if mod(k,5)==0                                                     %(6)
 fprintf('It=%3d |X-M|/|M|=%9.2e VioKKT=%9.2e\n',k,eps,VioKKT); end;%(7)
  k=k+1;      X0=X;      Y0=Y;                                      %(8)
  Y(Omega)=Y0(Omega)-(X0(Omega)-M(Omega))/s;                        %(9)
  A = X0 + Y/r;                    [U,D,V]=svd(A,0);               %(10)
  D=D-eye(n)/r;     D=max(D,0);      XT=(U*D)*V';        EX=X0-XT;  %(11)
  DXM=XT(Omega)-M(Omega);            eps = norm(DXM,'fro')/nM0;     %(12)
  YT(Omega)=Y0(Omega)-(XT(Omega)-M(Omega))/s;            EY=Y0-YT;  %(13)
  VioKKT =  max(max(abs(EX)))*r;                                   %(14)
   if (eps <= tol)   gamma=1;   end;                               %(15)
  X = X0 - EX*gamma;                                               %(16)
  Y(Omega) = Y0(Omega) - EY(Omega)*gamma;                          %(17)
end;                                                               %(18)
 fprintf('It=%3d  |X-M|/|M|=%9.2e  Vi0KKT=%9.2e \n',k,eps,VioKKT); %(19)
   RelEr=norm((X-M),'fro')/norm(M,'fro');   toc;                   %(20)
 fprintf(' Relative error = %9.2e  Rank(X)=%3d \n',RelEr,rank(X)); %(21)
 fprintf(' Violation of KKT Condition = %9.2e  \n',VioKKT);        %(22)
```



## 矩阵完整化问题: 用 Matlab-SVD 求解结果

| 1000×1000 matrix | ALM-based Classical PPA | | | | ALM-based Extended PPA | | | |
|---|---|---|---|---|---|---|---|---|
| (rank, $\frac{m}{d_r}$, $\frac{m}{n^2}$) | No. It. | CPU Sec. | Relative error | KKT-Violation | No. It. | CPU Sec. | Relative error | KKT-Violation |
| (10, 6, 0.12) | 85 | 989.84 | 9.62E-5 | 4.34E-6 | 77 | 899.40 | 9.32E-5 | 2.79E-6 |
| (50, 4, 0.39) | 43 | 491.35 | 1.46E-4 | 1.85E-5 | 37 | 425.24 | 1.21E-4 | 1.43E-5 |
| (100, 3, 0.58) | 36 | 394.68 | 1.72E-4 | 2.78E-5 | 31 | 363.04 | 1.55E-4 | 3.26E-5 |

♣ 用 Matlab 中的 SVD, 做一次 SVD 的花费很大, 总耗时与迭代次数成比例.

♣ 用 Extended PPA 与 Classical PPA 相比, 几乎不加额外的负担, 效率还是提高 $10\%$ 以上. 因此, 在任何情况下, 我们都提倡用 Extended PPA.

♣ 用 PROPACK [9] 中的 SVD, 快许多, 总耗时主要与问题性质有关. 由于我们主要对迭代次数感兴趣, 我们仅仅报道用 Matlab 做 SVD 的结果, 也附上所需要的子程序. 总耗时主要与问题性质有关.

♣ 注意到, Cai, Candès and Shen [2] 的方法对这三个例子的迭代次数分别是 117, 114 和129 (See the first three examples in Table 5.1 of [2], pp. 1974). 原则上, 每次迭代的主要工作量都是一次 SVD 分解. [2] 中采用不完全分解技术, 节省了总的运行时间.

**一点说明** 从第四讲到第七讲, 我们主要讲一类 Customized PPA 方法, 求解

$$\min\{\theta(x) \mid Ax = b, x \in \mathcal{X}\} \tag{5.11}$$

这样一类凸优化问题. 用来计算的例子, 都有应用背景. 用 Customized PPA 方法求解, 有很不错的数值结果. 主要原因是这些问题中的线性约束的矩阵 $A$ 的性态比较好 (注意到 diag$(X) = e$ 和 $X_\Pi = M_\Pi$ 表示成 $Ax = b$ 时, $A$ 只是一个投影矩阵), 加上这些问题中

$$\min\left\{\theta(x) + \frac{r}{2}\|x - a\|^2 \mid x \in \mathcal{X}\right\}$$

这样的子问题是有成熟方法求解的.

但是, 正如 R. Fletcher 在他的著作 Practical Methods of Optimization 中说到,

**Indeed there is no general agreement on the best approach and much research is still to be done.**

世界上没有一个方法是对所有的问题都是最好的, 我们不能指望用这几讲介绍的方法去求解形如 (5.11) 的一般凸优化问题, 特别是人为构造的难题. 好在一些有应用背景的问题, 常常有它的特殊结构, 简单的方法有时也能凑效.



# References


[1] D. P. Bertsekas. Constrained Optimization and Lagrange Multiplier Methods. Academic Press, 1982.

[2] J. F. Cai, E. J. Candès and Z. W. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim., **20**, 1956-1982, 2010.

[3] B.S. He, A class of projection and contraction methods for monotone variational inequalities, Applied Mathematics and Optimization, **35**, 69-76, 1997.

[4] B.S. He, Inexact implicit methods for monotone general variational inequalities, Math. Program. Series A **86**, 199-217, 1999.

[5] B. S. He, X. L. Fu and Z.K. Jiang, Proximal point algorithm using a linear proximal term, JOTA, **141**, 209-239, 2009.

[6] B.S He and L-Z Liao, Improvements of some projection methods for monotone nonlinear variational inequalities, JOTA, **112**, 111-128, 2002

[7] B. S. He and M.-H. Xu, A general framework of contraction methods for monotone variational inequalities, Pacific J. Optimization, **4**, 195-212, 2008.

[8] M. R. Hestenes, Multiplier and gradient methods. Journal of Optimization Theory and Applications,**4**, 302-320, 1969.

[9] R. M. Larsen, Propack – software for large and sparse svd calculation, http://sun.stanford.edu/rmunk/PROPACK/, 2005.

[10] J. Nocedal and S.J. Wright, Numerical Optimization, Springer Verlag, 1999.

[11] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, Optimization. Academic Press, 1969.

# 凸优化和单调变分不等式的收缩算法

## 第八讲: 基于梯度投影的凸优化
## 收缩算法和下降算法

Projected gradient-based contraction method
and descent method for convex optimization

南京大学数学系　何炳生
hebma@nju.edu.cn



**简单约束的凸优化问题**　这一讲讨论简单约束可微凸优化问题

$$\min \{f(x) \mid x \in \Omega\}$$

的梯度算法, 其中 $\Omega$ 是$\Re^n$ 中的凸闭集, 并假设到 $\Omega$ 上的投影是容易实现的. 在第一讲中就已经提到, 简单约束可微凸优化问题等价于求变分不等式

$$\text{VI}(\Omega, \nabla f) \qquad x \in \Omega, \qquad (x' - x)^T \nabla f(x) \geq 0, \quad \forall x' \in \Omega$$

的解. 这一讲的投影梯度方法, 分别是收缩算法和下降算法, 都不要用到函数值 $f(x)$, 只要对给定的 $x$, 能提供 $\nabla f(x)$. 收缩算法保证迭代点向解集靠近. 下降算法则隐含了目标函数值下降, 尽管目标函数值在计算过程中从不出现.

设 $x^*$ 是变分不等式 VI$(\Omega, \nabla f)$ 的解. 由于 $\tilde{x} = P_\Omega[x - \beta\nabla f(x)] \in \Omega$, 因此根据变分不等式的定义有第一个基本不等式

$$\boxed{\textbf{(FI1)} \quad (\tilde{x} - x^*)^T \beta \nabla f(x^*) \geq 0.}$$

由于 $\tilde{x}$ 是 $x - \beta\nabla f(x)$ 在 $\Omega$ 上的投影, $x^* \in \Omega$, 根据投影的基本性质, 有

$$\boxed{\textbf{(FI2)} \quad (\tilde{x} - x^*)^T \big([x - \beta\nabla f(x)] - \tilde{x}\big) \geq 0.}$$

# 1 凸优化的投影收缩算法

投影收缩算法由投影 $P_\Omega[x^k - \beta\nabla f(x^k)]$ 得到 $\tilde{x}^k$. $x^k \in \Omega^* \;\Leftrightarrow\; x^k = \tilde{x}^k$.

**误差度量函数** 一个非负函数 $\varphi(x^k, \tilde{x}^k)$ 称作变分不等式 VI$(\Omega, F)$ 的误差度量函数, 如果有 $\delta > 0$, 使得

$$\varphi(x^k, \tilde{x}^k) \ge \delta\|x^k - \tilde{x}^k\|^2, \quad \text{并且} \quad \varphi(x^k, \tilde{x}^k) = 0 \Leftrightarrow x^k = \tilde{x}^k. \tag{1.1}$$

**有利方向** 设矩阵 $G$ 正定, 凸优化的有利方向都是 $(x^k - \tilde{x}^k)$, 它满足

$$(x^k - x^*)^T G(x^k - \tilde{x}^k) \ge \varphi(x^k, \tilde{x}^k), \; \forall\, x^* \in \Omega^*. \tag{1.2}$$

**初等收缩算法** 考虑将 (1.1) 中的 $\varphi(x^k, \tilde{x}^k) \ge \delta\|x^k - \tilde{x}^k\|^2$ 改成条件

$$\varphi(x^k, \tilde{x}^k) \ge \frac{1}{2}\big(\|x^k - \tilde{x}^k\|_G^2 + \tau\|x^k - \tilde{x}^k\|^2\big), \quad (\tau > 0). \tag{1.3}$$

我们将条件 (1.2) 和 (1.3) 满足时, 用步长为 **1** 的迭代公式

$$x^{k+1} = x^k - (x^k - \tilde{x}^k) = \tilde{x}^k, \tag{1.4}$$



产生新迭代点的方法, 称为 **Primary Method** (初等方法).

## 1.1 凸二次优化的投影收缩算法

对凸二次优化, 我们将第一和第二个基本不等式

$$\begin{cases} (\tilde{x} - x^*)^T \beta(Hx^* + c) \ge 0 & (FI1) \\ (\tilde{x} - x^*)^T\big([x - \beta(Hx + c)] - \tilde{x}\big) \ge 0 & (FI2) \end{cases}$$

相加, 对所有的 $x \in \Re^n$, 都有

$$\{(x - x^*) - (x - \tilde{x})\}^T\{(x - \tilde{x}) - \beta H(x - x^*)\} \ge 0$$

由上式和 $H$ 半正定,

$$(x - x^*)^T(I + \beta H)(x - \tilde{x}) \ge \|x - \tilde{x}\|^2. \tag{1.1}$$

误差度量函数就是 $\varphi(x^k, \tilde{x}^k) = \|x^k - \tilde{x}^k\|^2$. 上式说明 $(\tilde{x}^k - x^k)$ 是未知距离函数 $\frac{1}{2}\|x - x^*\|_{(I+\beta H)}^2$ 在 $x^k$ 处的下降方向. 记 $G = I + \beta H$, 就是(1.2) 的形式

$$(x^k - x^*)^T G(x^k - \tilde{x}^k) \ge \varphi(x^k, \tilde{x}^k), \; \forall\, x^* \in \Omega^*.$$

这里考虑的投影收缩算法, 要求它产生的迭代点使得 $\|x^k - x^*\|_G^2$ 严格单调下降. 以

$$x(\alpha) = x^k - \alpha(x^k - \tilde{x}^k) \tag{1.2}$$

产生依赖于步长 $\alpha$ 的新迭代点. 考察与 $\alpha$ 相关的距离平方缩短量

$$\vartheta(\alpha) = \|x^k - x^*\|_G^2 - \|x(\alpha) - x^*\|_G^2. \tag{1.3}$$

利用 (1.1) 就有

$$\begin{aligned} \vartheta(\alpha) &= \|x^k - x^*\|_G^2 - \|x^k - x^* - \alpha(x^k - \tilde{x}^k)\|_G^2 \\ &\geq 2\alpha\|x^k - \tilde{x}^k\|^2 - \alpha^2\|x^k - \tilde{x}^k\|_G^2. \end{aligned} \tag{1.4}$$

我们得到 $\vartheta(\alpha)$ 的一个下界、二次函数 $q(\alpha)$,

$$q(\alpha) = 2\alpha\|x^k - \tilde{x}^k\|^2 - \alpha^2\|x^k - \tilde{x}^k\|_G^2. \tag{1.5}$$

使 $q(\alpha)$ 达到极大的

$$\alpha_k^* = \|x^k - \tilde{x}^k\|^2 / \|x^k - \tilde{x}^k\|_G^2. \tag{1.6}$$

用迭代式

$$x^{k+1} = x^k - \gamma\alpha_k^*(x^k - \tilde{x}^k), \quad \gamma \in (0,2) \tag{1.7}$$



产生新的迭代点 $x^{k+1}$. 这样的迭代序列 $\{x^k\}$ 满足

$$\|x^{k+1} - x^*\|_{(I+\beta H)}^2 \leq \|x^k - x^*\|_{(I+\beta H)}^2 - \gamma(2-\gamma)\alpha_k^*\|x^k - \tilde{x}^k\|^2. \tag{1.8}$$

换句话说, 迭代序列 $\{x^k\}$ 在 $G = (I + \beta H)$-模下向解集收缩. 关于凸二次优化在$(I + \beta H)$-模下收缩的算法, 更详细的可参见文献 [5]. 一般说来, 如何选取适当的 $\beta$ 对收敛速度影响很大.

**自调比投影收缩算法 $=$ 故意缩短了步长的最速下降法**

注意到在 $G = (I + \beta H)$ 时, “最优步长”

$$\alpha_k^* = \frac{\|x^k - \tilde{x}^k\|^2}{(x^k - \tilde{x}^k)^T(I + \beta H)(x^k - \tilde{x}^k)}. \tag{1.9}$$

对给定的 $\nu \in (0,1)$, 我们使用自调比法则选取 $\beta$ 使得

$$(x^k - \tilde{x}^k)^T(\beta H)(x^k - \tilde{x}^k) \leq \nu\|x^k - \tilde{x}^k\|^2, \tag{1.10}$$

代入 (1.9) 便有

$$\alpha_k^* \geq \frac{1}{1+\nu} > \frac{1}{2}.$$

这使得我们有可能在 (1.7) 中动态地取

$$\gamma_k = 1/\alpha_k^*, \qquad \text{因此} \qquad 1 < \gamma_k \le 1+\nu < 2.$$

迭代式 (1.7) 就成为

$$x^{k+1} = \tilde{x}^k = P_\Omega[x^k - \beta(Hx^k + c)]. \tag{1.11}$$

这样做使得有可能每步迭代只计算一次目标函数的梯度(矩阵与向量相乘).

在 (1.8) 中利用 $\gamma_k\alpha_k^* = 1,\ \gamma_k \le 1+\nu$ 以及 $\tilde{x}^k = x^{k+1}$ 就有

$$\|x^{k+1} - x^*\|^2_{(I+\beta H)} \le \|x^k - x^*\|^2_{(I+\beta H)} - (1-\nu)\|x^k - x^{k+1}\|^2. \tag{1.12}$$

♣ 求解 $\min\{\frac{1}{2}x^T Hx + c^T x\}$ 的最速下降法是

$$x^{k+1} = x^k - \alpha_k^{SD}(Hx^k + c), \qquad \alpha_k^{SD} = \frac{\|Hx^k + c\|^2}{(Hx^k + c)^T H(Hx^k + c)}.$$

当 $\Omega = \Re^n$ 时, 根据 (1.11) 和 (1.10) 有

$$x^{k+1} = x^k - \beta(Hx^k + c) \qquad \text{和} \qquad \beta \le \nu\alpha_k^{SD}.$$



因此, [5] 中符合条件 (1.10) 的投影收缩算法(1.11) 相当于将最速下降法故意缩短了步长. 换句话说, 据此设计的求解无约束凸二次优化的算法, 恰是缩小了步长的最速下降法. 我们曾担心用这些方法求解无约束凸二次规划效果会比最速下降法还差. 事实上, 对最速下降法故意缩短步长, 收敛速率有令人难以置信的数量级提高. 读者容易用下面的例子来验证.

**数值试验** 试验问题中的 Hessian 矩阵是 Hilbert 矩阵, 即：

$$H = \{h_{ij}\}, \quad h_{ij} = \frac{1}{i+j-1}, \quad i = 1,\cdots,n; \quad j = 1,\cdots,n.$$

问题的规模(维数)分别从从 $100$ 到 $500$. 因为 Hilbert 矩阵的条件很坏, 我们将最优解 $x^*$ 设定为每个分量都是 1 的向量, 然后令 $c = -Hx^*$, 再用梯度类算法去求解. 试验中, 我们分别将零向量、 $c$ 和 $-c$ 取作初始向量. 这里采用的停机准则是$\|Hx^k + c\|/\|Hx^0 + c\| \le 10^{-7}$. 表 1–3 分别列出了在不同问题规模、不同缩扩(缩减或扩张)因子和不同初始向量下的迭代次数. 其中 $n$ 表示问题规模, $r$ 表示缩扩因子. $r = 1$ 时, 就是最速下降法.

初步的试验结果不但解除了我们的上述担心, 同时也印证了 Dai 和 Yuan [2] 提到的将最速下降法的步长乘上一个小于 $1$ 的因子会加快收敛的结论.

表 1. 初始向量 $x^0 = 0$, 使用不同缩扩因子 $r$ 时的迭代次数

| n= | 0.1 | 0.3 | 0.5 | 0.7 | 0.8 | **0.9** | 0.95 | 0.99 | **1.00** | 1.20 |
|---|---|---|---|---|---|---|---|---|---|---|
| 100 | 2863 | 1346 | 853 | 627 | 582 | **437** | 565 | 1201 | **13169** | 22695 |
| 200 | 3283 | 1398 | 923 | 804 | 541 | **669** | 898 | 1178 | **14655** | 21083 |
| 300 | 3497 | 1323 | 856 | 739 | 720 | **568** | 619 | 1545 | **17467** | 24027 |
| 500 | 3642 | 1351 | 1023 | 773 | 667 | **578** | 836 | 2024 | **17757** | 22750 |

初始向量 $x^0 = 0$, 迭代结束时平均相对误差 $\|x^k - x^*\|/\|x^0 - x^*\| = 3.0\text{e}-3$

表 2. 初始向量 $x^0 = c$, 使用不同缩扩因子 $r$ 时的迭代次数

| n= | 0.1 | 0.3 | 0.5 | 0.7 | 0.8 | **0.9** | 0.95 | 0.99 | **1.0** | 1.2 |
|---|---|---|---|---|---|---|---|---|---|---|
| 100 | 2129 | 1034 | 544 | 424 | 302 | **438** | 568 | 919 | **5527** | 9667 |
| 200 | 1880 | 808 | 568 | 482 | 372 | **339** | 446 | 713 | **6625** | 11023 |
| 300 | 1852 | 1002 | 741 | 531 | 610 | **452** | 450 | 917 | **6631** | 10235 |
| 500 | 2059 | 939 | 568 | 573 | 379 | **547** | 558 | 874 | **7739** | 11269 |

初始向量 $x^0 = c$, 迭代结束时平均相对误差 $\|x^k - x^*\|/\|x^0 - x^*\| = 1.8\text{e}-3$

表 3. 初始向量 $x^0 = -c$, 使用不同缩扩因子 $r$ 时的迭代次数

| n= | 0.1 | 0.3 | 0.5 | 0.7 | 0.8 | **0.9** | 0.95 | 0.99 | **1.0** | 1.2 |
|---|---|---|---|---|---|---|---|---|---|---|
| 100 | 2545 | 1221 | 666 | 591 | 498 | **482** | 638 | 1581 | **14442** | 20380 |
| 200 | 2826 | 990 | 874 | 470 | 526 | **455** | 578 | 841 | **15222** | 18892 |
| 300 | 2891 | 1299 | 918 | 738 | 549 | **571** | 608 | 2552 | **18762** | 21208 |
| 500 | 3158 | 1769 | 909 | 678 | 506 | **512** | 678 | 1240 | **17512** | 19790 |

初始向量 $x^0 = -c$, 迭代结束时平均相对误差 $\|x^k - x^*\|/\|x^0 - x^*\| = 3.8\text{e}-3$



## 1.2 基于 FI2 的非线性凸优化的投影收缩算法

可微凸优化问题 $\min\{f(x)\,|\,x \in \Omega\}$ 与变分不等式 VI$(\Omega, \nabla f)$ 等价. 由

$$\tilde{x} = P_\Omega[x - \beta\nabla f(x)]$$

和投影的基本性质, 可以得到

$$(\tilde{x} - x^*)^T\big([x - \beta\nabla f(x)] - \tilde{x}\big) \geq 0.$$

上面的不等式其实就是第二讲中的第二个基本不等式. 据此可以推得

$$\begin{aligned}(\tilde{x} - x^*)^T(x - \tilde{x}) &\geq (\tilde{x} - x^*)^T\beta\nabla f(x) \\ &= (x - x^*)^T\beta\nabla f(x) - (x - \tilde{x})^T\beta\nabla f(x). \end{aligned} \tag{1.13}$$

另据凸函数的性质, 有

$$(x - x^*)^T\nabla f(x) \geq f(x) - f(x^*) \geq (x - \tilde{x})^T\nabla f(\tilde{x}) + (f(\tilde{x}) - f(x^*)). \tag{1.14}$$

以 (1.14) 代入 (1.13) 的右端, 则对所有的 $x \in \Re^n$, 都有

$$(\tilde{x} - x^*)^T(x - \tilde{x}) \geq -\beta(x - \tilde{x})^T(\nabla f(x) - \nabla f(\tilde{x})).$$

由上式得

$$(x-x^*)^T(x-\tilde{x}) \geq \|x-\tilde{x}\|^2 - \beta(x-\tilde{x})^T(\nabla f(x)-\nabla f(\tilde{x})). \tag{1.15}$$

我们定义

$$\varphi(x,\tilde{x}) = \|x-\tilde{x}\|^2 - \beta(x-\tilde{x})^T(\nabla f(x)-\nabla f(\tilde{x})). \tag{1.16}$$

假设 $\nabla f$ 是 Lipschitz 连续的. 对于一个确定的 $\nu \in (0,1)$, 总可以采用 Armijo 技术对算子进行调比, 使得 $\beta\nabla f$ 的 Lipschitz 常数不大于 $\nu$, 有

$$\beta\|\nabla f(x)-\nabla f(\tilde{x})\| \leq \nu\|x-\tilde{x}\|$$

式成立. 这样便有

$$(x-x^*)^T(x-\tilde{x}) \geq \varphi(x,\tilde{x}) \geq (1-\nu)\|x-\tilde{x}\|^2.$$

由 (1.16) 定义的 $\varphi(x,\tilde{x})$ 满足条件 (1.1), 其中的 $\delta = 1-\nu > 0$. 考虑欧氏模下的收缩算法, 由

$$x^{k+1} = x^k - \gamma\alpha_k^*(x^k - \tilde{x}^k),$$



产生新的迭代点 $x^{k+1}$, 其中

$$\alpha_k^* = \frac{\varphi(x^k,\tilde{x}^k)}{\|x^k-\tilde{x}^k\|^2}, \qquad \gamma \in (0,2).$$

迭代序列 $\{x^k\}$ 就满足

$$\|x^{k+1}-x^*\|^2 \leq \|x^k-x^*\|^2 - \gamma(2-\gamma)\alpha_k^*\varphi(x^k,\tilde{x}^k).$$

由 $\varphi(x,\tilde{x}) \geq (1-\nu)\|x-\tilde{x}\|^2$ 可以进一步推得

$$\|x^{k+1}-x^*\|^2 \leq \|x^k-x^*\|^2 - \gamma(2-\gamma)(1-\nu)^2\|x^k-\tilde{x}^k\|^2.$$

有关这一节的相关论文和数值试验可见:


- B.S. He, L.Z. Liao, and X. Wang, Proximal-like contraction methods for monotone variational inequalities in a unified framework I: Effective quadruplet and primary methods, Comput. Optim. Appl., 51, 649-679, 2012
- B.S. He, L.Z. Liao, and X. Wang, Proximal-like contraction methods for monotone variational inequalities in a unified framework II: General methods and numerical experiments, Comput. Optim. Appl. 51, 681-708, 2012

## 2 基于 FI2 的凸优化投影梯度下降算法

设 $\Omega \subset \Re^n$ 是闭凸集, 考虑下面的变分不等式

$$x^* \in \Omega, \quad (x - x^*)^T g(x^*) \geq 0, \qquad \forall\, x \in \Omega, \tag{2.1}$$

其中 $g(x) : \Re^n \to \Re^n$. 设 $g(x)$ 是某个凸函数 $f(x)$ 的梯度, 然而 $f(x)$ 是无法提供的. 只是对给定的 $x$, 可以观测到 $g(x)$, 而这种观测往往是代价不菲的. 因此, (2.1) 形式上等价于约束凸优化问题

$$\min \{f(x) \mid x \in \Omega\}. \tag{2.2}$$

我们称 (2.2) 为**Oracle** 凸优化问题, 求解 (2.2), 只有 $g(x)$ 的信息可以使用. 我们假设 (2.2) 的解集非空并用 $\Omega^*$ 表示, 对解点 $x^* \in \Omega^*$, 假设 $f(x^*) > -\infty$. 此外还假设 $g(x)$ 是 Lipschitz 连续的, 也就是说, 存在常数 $L > 0$ 使得

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall\, x, y \in \Re^n.$$

这一节的算法, 并不涉及函数 $f(x)$, 却隐含了函数值 $f(x^k)$ 严格单调下降, 因此属于下降算法.



**单步投影梯度法** (Single step projected gradient method)

Step 0. Take $x^0 \in \Omega$.

Step $k$. ($k \geq 0$) Set

$$x^{k+1} = P_\Omega[x^k - \beta_k g(x^k)], \tag{2.3a}$$

where the step size $\beta_k$ satisfies the following condition:

$$(x^k - x^{k+1})^T (g(x^k) - g(x^{k+1})) \leq \frac{\nu}{\beta_k} \|x^k - x^{k+1}\|^2. \tag{2.3b}$$

这里说的单步投影梯度法只生成一个序列 $\{x^k\}$, 它有别于 Nesterov [10] 的除了生成 $\{x^k\}$ 以外, 同时生成一个辅助序列 $\{y^k\}$ 的加速方法. 关于只用梯度的加速方法, 我们在这一讲的最后一节介绍.

注意到, 当 $\beta_k \leq \nu/L$, 其中 $L$ 是 $g(x)$ 的Lipschitz 常数, 条件 (2.3b) 就满足, 因为:

$$\begin{aligned}
&(x^k - x^{k+1})^T \beta_k (g(x^k) - g(x^{k+1})) \\
&\leq \quad \|x^k - x^{k+1}\| \cdot \beta_k L \|x^k - x^{k+1}\| \\
&\leq \quad \nu \|x^k - x^{k+1}\|^2.
\end{aligned}$$

我们首先利用投影和凸函数的基本性质, 证明一个重要的引理.

**Lemma 2.1** *对给定的* $x^k$, *设* $x^{k+1}$ *由* (2.3a) *给出. 如果步长* $\beta_k$ *满足准则* (2.3b), *则有*

$$(x-x^{k+1})^T g(x^k) \ge \frac{1}{\beta_k}(x-x^{k+1})^T(x^k-x^{k+1}), \quad \forall x\in\Omega, \tag{2.4}$$

*和*

$$\begin{aligned}&\beta_k(f(x)-f(x^{k+1}))\\ &\quad\ge \quad (x-x^{k+1})^T(x^k-x^{k+1})-\nu\|x^k-x^{k+1}\|^2, \ \forall\, x\in\Omega.\end{aligned} \tag{2.5}$$

**证明**. $x^{k+1}$ 是 $[x^k-\beta_k g(x^k)]$ 在 $\Omega$ 上的投影 (see (2.3a)), 根据投影性质有

$$(x-x^{k+1})^T\{[x^k-\beta_k g(x^k)]-x^{k+1}\}\le 0, \quad \forall\, x\in\Omega,$$

由此得到

$$(x-x^{k+1})^T\beta_k g(x^k) \ge (x-x^{k+1})^T(x^k-x^{k+1}), \quad \forall\, x\in\Omega. \tag{2.6}$$

引理的第一个结论 (2.4) 得证. 利用 $f$ 的凸性, 我们分别有

$$f(x) \quad\ge\quad f(x^k)+(x-x^k)^T g(x^k). \tag{2.7}$$



和

$$\begin{aligned} f(x^{k+1}) &\le f(x^k)+(x^{k+1}-x^k)^T g(x^{k+1})\\ &= f(x^k)+(x^{k+1}-x^k)^T g(x^k)+(x^{k+1}-x^k)^T\big(g(x^{k+1})-g(x^k)\big)\\ &\le f(x^k)+(x^{k+1}-x^k)^T g(x^k)+\frac{\nu}{\beta_k}\|x^k-x^{k+1}\|^2. \end{aligned} \tag{2.8}$$

以上最后一步的 “$\le$” 用到了条件 (2.3b). 由 (2.7) 和 (2.8) 得到

$$\begin{aligned} &f(x)-f(x^{k+1})\\ &\quad\ge f(x^k)+(x-x^k)^T g(x^k)\\ &\qquad -\Big\{f(x^k)+(x^{k+1}-x^k)^T g(x^k)+\frac{\nu}{\beta_k}\|x^k-x^{k+1}\|^2\Big\}\\ &\quad= (x-x^{k+1})^T g(x^k)-\frac{\nu}{\beta_k}\|x^k-x^{k+1}\|^2. \end{aligned} \tag{2.9}$$

将结论 (2.4) 代入 (2.9), 我们得到

$$f(x)-f(x^{k+1}) \ge \frac{1}{\beta_k}(x-x^{k+1})^T(x^k-x^{k+1})-\frac{\nu}{\beta_k}\|x^k-x^{k+1}\|^2,$$

引理的第二部分得证. $\Box$

下面的定理说明单步投影梯度法 (2.3) 是保证函数值严格单调下降的算法.

**Theorem 2.1** *设序列 $\{x^k\}$ 由单步投影梯度法 (2.3) 给出. 则有*

$$f(x^{k+1}) \le f(x^k) - \frac{1-\nu}{\beta_k}\|x^k - x^{k+1}\|^2, \tag{2.10}$$

*and*

$$\begin{aligned} \|x^{k+1} - x^*\|^2 \quad \le \quad & \|x^k - x^*\|^2 - (1-2\nu)\|x^k - x^{k+1}\|^2 \\ & -2\beta_k(f(x^{k+1}) - f(x^*)). \end{aligned} \tag{2.11}$$

**Proof**. 在引理 2.1 的 (2.5) 中令 $x = x^k$, 便证得定理得第一个结论. 以 $x = x^*$ 代入 (2.5) 中就有

$$\beta_k(f(x^*) - f(x^{k+1})) \ge (x^* - x^{k+1})^T(x^k - x^{k+1}) - \nu\|x^k - x^{k+1}\|^2,$$

并因此有

$$(x^k - x^*)^T(x^k - x^{k+1}) \ge (1-\nu)\|x^k - x^{k+1}\|^2 + \beta_k(f(x^{k+1}) - f(x^*)).$$



根据上式, 我们得到

$$\begin{aligned} \|x^{k+1} - x^*\|^2 \quad &= \quad \|(x^k - x^*) - (x^k - x^{k+1})\|^2 \\ &= \quad \|x^k - x^*\|^2 - 2(x^k - x^*)^T(x^k - x^{k+1}) + \|x^k - x^{k+1}\|^2 \\ &\le \quad \|x^k - x^*\|^2 - (1-2\nu)\|x^k - x^{k+1}\|^2 \\ & \quad\quad -2\beta_k(f(x^{k+1}) - f(x^*)). \end{aligned}$$

定理得第二个结论得证. □

**Theorem 2.2** *设序列 $\{x^k\}$ 由单步投影梯度法 (2.3) 给出.*

1. *如果 $\nu < 1$ 和 $\inf\{\beta_k\} = \beta_{\min} > 0$, 则有*

$$\lim_{k\to\infty}\left(f(x^k) - f(x^*)\right) = 0.$$

2. *进一步, 如果 $\nu \le \frac{1}{2}$, 则更有*

$$\|x^{k+1} - x^*\|^2 \le \|x^k - x^*\|^2 - 2\beta_k(f(x^{k+1}) - f(x^*)),$$

   *因此, 生成的序列 $\{x^k\}$ 对于解集 $\Omega^*$ 是 Fejér 单调的.*

**Proof**. 可以直接从定理 2.1 的结论推得. □

以下证明单步投影梯度法 (2.3) 的迭代复杂性是 $O(1/k)$. 为了形式上的方便, 我们假设 $\beta_k \equiv \beta$.

**Theorem 2.3** *Let $\{x^k\}$ be generated by the single step projected gradient method. Then, we have*

$$
\begin{aligned}
& 2k\beta(f(x^*) - f(x^k)) \\
& \geq \quad \sum_{l=0}^{k-1}\Big((1-2\nu) + 2l(1-\nu)\Big)\|x^l - x^{l+1}\|^2 - \|x^0 - x^*\|^2. \qquad (2.12)
\end{aligned}
$$

**Proof**. First, it follows from (2.11) that, for any $x^* \in \Omega^*$ and all $l \geq 0$, we have

$$2\beta(f(x^*) - f(x^{l+1})) \geq \|x^{l+1} - x^*\|^2 - \|x^l - x^*\|^2 + (1-2\nu)\|x^l - x^{l+1}\|^2.$$

Summing the above inequality over $l = 0, \ldots, k-1$, we obtain

$$
\begin{aligned}
& 2\beta\Big(kf(x^*) - \sum_{l=0}^{k-1} f(x^{l+1})\Big) \\
& \geq \quad \|x^k - x^*\|^2 - \|x^0 - x^*\|^2 + \sum_{l=0}^{k-1}(1-2\nu)\|x^l - x^{l+1}\|^2. \qquad (2.13)
\end{aligned}
$$



It follows from (2.10) that

$$2\beta\big(f(x^l) - f(x^{l+1})\big) \geq 2(1-\nu)\|x^l - x^{l+1}\|^2.$$

Multiplying the last inequality by $l$ and summing over $l = 0, \ldots, k-1$, it follows that

$$2\beta\sum_{l=0}^{k-1}\Big(lf(x^l) - (l+1)f(x^{l+1}) + f(x^{l+1})\Big) \geq \sum_{l=0}^{k-1} 2l(1-\nu)\|x^l - x^{l+1}\|^2,$$

which simplifies to

$$2\beta\Big(-kf(x^k) + \sum_{l=0}^{k-1} f(x^{l+1})\Big) \geq \sum_{l=0}^{k-1} 2l(1-\nu)\|x^l - x^{l+1}\|^2. \qquad (2.14)$$

Adding (2.13) and (2.14) and ignoring the positive term $\|x^k - x^*\|^2$, we get

$$
\begin{aligned}
2k\beta\big(f(x^*) - f(x^k)\big) \quad & \geq \quad \|x^k - x^*\|^2 - \|x^0 - x^*\|^2 \\
& \qquad + \sum_{l=0}^{k-1}\Big((1-2\nu) + 2l(1-\nu)\Big)\|x^l - x^{l+1}\|^2,
\end{aligned}
$$

which implies (2.12) and the theorem is proved. $\Box$

**Theorem 2.4** *Let $\{x^k\}$ be generated by the single step projected gradient method. If $\nu \le \frac{1}{2}$, then we have*

$$f(x^k) - f(x^*) \le \frac{\|x^0 - x^*\|^2}{2k\beta}, \tag{2.15}$$

*and thus the iteration-complexity of this method is $O(1/k)$.*

**Proof**. For $\nu \le \frac{1}{2}$ and $l \ge 0$, $(1-2\nu) + 2l(1-\nu) > 0$. It follows from (2.12) that

$$2k\beta(f(x^*) - f(x^k)) \ge -\|x^0 - x^*\|^2,$$

which implies (2.15) and the assertion is proved. $\square$

♣ In fact, for any $\nu \in (0,1)$, the iteration-complexity of the single projected gradient method is $O(1/k)$. For example, if $\nu = 0.9$, then

$$(1-2\nu) + 2l(1-\nu) \ge 0, \ \forall\, l \ge 4.$$

In this case, it follows from (2.12) that

$$2k\beta(f(x^*) - f(x^k)) \ge \sum_{l=0}^{3}\Big((1-2\nu) + 2l(1-\nu)\Big)\|x^l - x^{l+1}\|^2 - \|x^0 - x^*\|^2.$$



and thus

$$\begin{aligned} & 2k\beta(f(x^k) - f(x^*)) \\ \le \ & \|x^0 - x^*\|^2 + \sum_{l=0}^{3}\Big((2\nu-1) + 2l(\nu-1)\Big)\|x^l - x^{l+1}\|^2. \end{aligned} \tag{2.16}$$

Since $\nu \le 0.9$, we have

$$\begin{aligned} & 2k\beta(f(x^k) - f(x^*)) \\ \le \ & \|x^0 - x^*\|^2 + \frac{4}{5}\|x^0 - x^1\|^2 + \frac{3}{5}\|x^1 - x^2\|^2 \\ & + \frac{2}{5}\|x^2 - x^3\|^2 + \frac{1}{5}\|x^3 - x^4\|^2. \end{aligned} \tag{2.17}$$

Consequently

$$f(x^k) - f(x^*) \le \frac{1}{2k\beta}\Big(\|x^0 - x^*\|^2 + \frac{4}{5}\sum_{l=0}^{3}\|x^l - x^{l+1}\|^2\Big). \tag{2.18}$$

在实际计算中, 我们不是用 (2.3), 而建议采用以下自调比的单步投影梯度法.

自调比的单步投影梯度法:

Set $\beta_0 = 1$, $\mu = 0.5$, $\nu = 0.9$, $x^0 \in \Omega$ and $k = 0$. Provide $g(x^0)$.

For $k = 0, 1, \ldots$, if the stopping criterium is not satisfied, do

Step 1. $\tilde{x}^k = P_\Omega[x^k - \beta_k g(x^k)]$,

$r_k = \beta_k \|g(x^k) - g(\tilde{x}^k)\| / \|x^k - \tilde{x}^k\|$

**while** $r_k > \nu$

$\beta_k := \beta_k * 0.8 / r_k$,

$\tilde{x}^k = P_\Omega[x^k - \beta_k g(x^k)]$,

$r_k = \beta_k \|(g(x^k) - g(\tilde{x}^k)\| / \|x^k - \tilde{x}^k\|$.

**end(while)**

$x^{k+1} = \tilde{x}^k$,

$g(x^{k+1}) = g(\tilde{x}^k)$.

**If** $r_k \le \mu$ then $\beta_k := \beta_k * 1.5$, **end(if)**

Step 2. $\beta_{k+1} = \beta_k$ and $k = k + 1$, go to Step 1.

我们称之为 single step projected gradient method, 因为在不需要调整 $\beta_k$ 时, 每次迭代只需要提供一次梯度值.



# 3 经济平衡问题上的一个应用

作为单步投影梯度法的应用, 我们讨论第一讲 §2 提到的保护资源、保障供给互补问题的求解方法. 读者可以从第一讲的 §2 中更好地了解问题的背景. 对所有的 $i = 1, \ldots, m$, $j = 1, \ldots, n$, 使用记号:

- $S_i$: 该种商品的第 $i$ 个资源地;
- $D_j$: 该种商品的第 $j$ 个需求地;
- $x_{ij}$: 从 $S_i$ 到 $D_j$ 的交易量;
- $s_i$: 经营者们在资源地 $S_i$ 的总采购量, $s_i = \sum_{j=1}^n x_{ij}$;
- $d_j$: 经营者在需求地 $D_j$ 的总销售量, $d_j = \sum_{i=1}^m x_{ij}$;
- $h_i^s$: 经营者在资源地 $S_i$ 处的采购价;
- $h_j^d$: 经营者在需求地 $D_j$ 处的销售价;
- $t_{ij}$: 从 $S_i$ 到 $D_j$ 的交易费用(包括运输费用);
- $y_i$: 政府为避免资源过度开采而在资源地 $S_i$ 向经营者征收的资源税;
- $z_j$: 政府为保障供给而在需求地 $D_j$ 给经营者的经营补贴.

### 3.1 经营者追求利益最大化之互补问题

显然, 为了获利, 如果 $S_i$ 处的采购价、资源税及从 $S_i$ 到 $D_j$ 的交易费的和 $(h_i^s + y_i + t_{ij})$ 不小于需求地 $D_j$ 处的销售价与政府补贴的和 $(h_j^d + z_j)$, 经营者是不会从 $S_i$ 采购商品运到 $D_j$ 销售的. 反之, 根据贪婪原理, 经营者会尽可能增大经营量, 直到 $(h_i^s + y_i + t_{ij})$ 和 $(h_j^d + z_j)$ 相等. 这种关系的数学表达式是下面的平衡问题:

$$h_i^s + y_i + t_{ij} \begin{cases} \geq h_j^d + z_j, & \text{if } x_{ij} = 0, \\ = h_j^d + z_j, & \text{if } x_{ij} > 0\,. \end{cases} \tag{3.1}$$

它可以写成互补问题的形式: 对任意的 $i = 1, \ldots, m$ 和 $j = 1, \ldots, n$, 都有

$$0 \leq x_{ij} \;\perp\; \big((h_i^s + y_i + t_{ij}) - (h_j^d + z_j)\big) \geq 0. \tag{3.2}$$

通常假设 $S_i$ 处的采购价 $f_i$ 仅与经营者们在该地的采购量 $s_i$ 有关; $D_j$ 处的销售价 $g_j$ 仅与经营者们运到该地的销售量 $d_j$ 有关; 从 $S_i$ 到 $D_j$ 的交易价仅与它们之间的交易量 $x_{ij}$有关. 对于 $y = 0$ 和 $z = 0$ 的互补问题 (3.2), 文献中 [12] 称之为空间价格平衡问题 (Spatial Price Equilibrium Problem). 已有的求解空



间价格平衡问题的方法[?, 8, 9] 都对函数 $h^s$, $h^d$ 和 $t$ 的表达式有一定的要求. 当 $a_i$, $b_j$, $c_{ij}$, $\xi_i$, $\eta_j$, $\zeta_{ij}$ 为常数,

$$h_i^s(s_i) = \xi_i + a_i s_i, \qquad a_i \geq 0; \tag{3.3a}$$

$$h_j^d(d_j) = \eta_j - b_j d_j, \qquad b_j \geq 0; \tag{3.3b}$$

$$t_{ij}(x_{ij}) = \zeta_{ij} + c_{ij} x_{ij}, \qquad c_{ij} \geq 0 \tag{3.3c}$$

时, (3.2) 就是一个单调对称线性互补问题, 可以用求解带非负约束凸二次规划的方法求解 [3]. 在实际生活中, 这个问题是由市场根据贪婪原理自行解决的.

### 3.2 保护资源和保障供给的经济平衡问题

在现实生活中, 对(3.2) 中给定的 $y \geq 0$ 和 $z \geq 0$, 经济规律这只 “无形的手” 会让经营者们根据贪婪原理找到相应的 $(x, s, d)$, 它是变分不等式(3.2) 的解. 换句话说, 原问题的解可由经营者们在经营活动中自行给出. 我们通常只知道函数 $f$, $g$ 和 $t$ 的一些性质而不知道它们的具体表达式, 能够观察到的是这个依赖于给定 $y$ 和 $z$ 的 $s$ 与 $d$, 由于它是经营者根据贪婪原理给出的, 因此不可能顾及保护资源和保障供给. 假设从可持续发展的要求允许的最大资源消耗

量是$s^{\max}$, 为保障供给而必需的最小供应量是$d^{\min}$, 它们之间满足相容关系

$$\sum_{i=1}^{m} s_i^{\max} \geq \sum_{j=1}^{n} d_j^{\min}.$$

职能部门要采取经济手段来保证

$$s \leq s^{\max} \quad 和 \quad d \geq d^{\min}. \tag{3.4}$$

具体说来，在过度热销的资源地 $S_i$ 通过对经营者征收资源税 $y_i$ 保护资源，对供应不足的需求地 $D_j$ 通过给经营者补贴 $z_j$ 而吸引经营者增加供给. 我们的任务则是用数学方法帮助职能部门找到最优税收向量 $y^* \in \Re_+^m$ 和最优补贴向量 $z^* \in \Re_+^n$, 使得对我们给出的 $(y^*, z^*)$ 以及经营者们由此产生的空间价格平衡问题 (3.2) 的解中的 $s^*$ 和 $d^*$ 满足

$$y^* \geq 0, \quad s^{\max} - s^* \geq 0, \quad {y^*}^T (s^{\max} - s^*) = 0, \tag{3.5}$$

和

$$z^* \geq 0, \quad d^* - d^{\min} \geq 0, \quad {z^*}^T (d^* - d^{\min}) = 0. \tag{3.6}$$



记

$$u = \begin{pmatrix} y \\ z \end{pmatrix} \quad 和 \quad F(u) = \begin{pmatrix} s^{\max} - s(u) \\ d(u) - d^{\min} \end{pmatrix}. \tag{3.7}$$

根据以上分析, 我们的任务可以归结为求解以下的*隐式互补问题*

$$u \geq 0, \quad F(u) \geq 0, \quad u^T F(u) = 0. \tag{3.8}$$

这里我们所说的'*隐式*' 是指我们不知道函数 $F$ 的显式表达式而只能对给定的 $u \geq 0$ 观察到 $F(u)$ 的值. 本文的隐式互补问题是带有附加约束 (3.4) 的空间价格平衡问题 (3.2) 的对偶问题. 下面我们对函数 $t_{ij}, h_i^s$ 和 $h_j^d$ 作一定的假设后讨论隐式互补问题 (3.8) 的性质.

**假设**

A1. 对任意的 $i = 1, \ldots, m$ 和 $j = 1, \ldots, n$, 交易费用 $t_{ij}(x_{ij})$ 是交易量 $x_{ij}$ 的非减函数.

A2. $h_i^s$ 和 $h_j^d$ 分别是 $s_i$ 和 $d_j$ 的一致严格增和一致严格减函数.

这样的假设应该说是合理的. 原因是由于道路拥挤, 单位交易费用(其中大部分为运输费用) 不会因交易量增加而减小, 资源地的采购价会因“采购量”

的增大(货俏)而被生产者提高, 需求地的销售价会随“到货量”的增加而降低. 由以上的假设, 隐式互补问题 (3.8) 中的向量函数 $F(u)$ 是单调和 Lipschitz 连续的.

## 3.3 求解隐式互补问题的直接迭代方法

对映射 $F$ 是单调和 Lipschitz 连续的隐式互补问题 (3.8), 文献中已有的在迭代过程中只用到 $F(u)$ 的值的方法, 主要是外梯度法 (Extra-gradient Method) 和投影收缩算法 (Projection and Contraction Method). 我们分别在第二讲和第三讲中已经做了介绍. 对于这一节讨论的隐式互补问题, 实际问题也只为我们提供了对应于自变量 $u \geq 0$ 的函数值 $F(u)$ 这样的信息.

由于每调用一次函数值就等同要进行一次税收和补贴政策的调整，实际问题要求我们在求解过程中尽可能减少调用函数值的次数.

♣ 注意到预测过程中至少调用二次函数值, 分别调用 $F(u^k)$ 和 $F(\tilde{u}^k)$ 以观察 $\tilde{u}^k$ 能否被接受为预测点. 在 $\beta_k$ 选取适当, 进行了一次试探 $\tilde{u}^k$ 就被接受为预测点的情况时则预测过程恰好调用了二次函数值.



## 3.4 初步的数值试验情况

我们感兴趣的是用 §4 中介绍的单步投影梯度法, 求解保护资源和保障供给的经济平衡问题 (3.8). 为此, 对给定的 $y \in R^m_+$ 和 $z \in R^n_+$, 经营者们要解一个原问题 (3.2). 我们对原问题 (3.2), 按照 (3.3) 中的方式定义函数 $h^s$, $h^d$ 和 $t$. 设 $m = 20,\ n = 50$. 对 $i = 1, \ldots, m$ 和$j = 1, \ldots, n$, 取

$$\begin{array}{lll} a_i \in (1,2), & b_j \in (1,2), & c_{ij} \in (0.002, 0.005), \\ \xi_i \in (300,400), & \eta_j \in (600,700), & \zeta_{ij} \in (10,20), \end{array}$$

为一定范围中的随机数. 这样, 对给定的 $y$ 和 $z$, 经营者们解的原问题 (3.2) 是一个单调对称的线性互补问题, 相当于变量个数为 $1000$ 的带非负约束的凸二次规划. 这个本该由经营者们求解的问题, 也可用 §3 中介绍的方法去求解.

为保护资源和保障供给, 分别取 $s^{\max}$ 和 $d^{\min}$ 的每个分量均为 $150$ 和 $40$. 我们要求解的隐式对偶问题的变量个数是 $70$. 计算以 $u^0 = 0$ 为初始点. 由于求解互补问题 (3.8) 时可以将 $\|\min\{u^k, F(u^k)\}\|_\infty$ 作为误差的一种度量, 我们在下表中给出对精度 $\|\min\{u^k, F(u^k)\}\|_\infty$ 不同要求时, 不同方法所需要的迭代次数和调用函数值 $F(u)$ 的次数.

不同方法达到同样精度要求的迭代次数和调用 $F(u)$ 的次数

| 误差精度 | 外梯度方法 | | 投影收缩算法D1 | | 投影收缩算法D2 | | 投影梯度法 | |
|---|---|---|---|---|---|---|---|---|
| $\|e(u)\|_\infty$ | 迭代次数 | 调用梯度次数 | 迭代次数. | 调用梯度次数 | 迭代次数 | 调用梯度次数 | 迭代次数 | 调用梯度次数 |
| 1 | 35 | 72 | 5 | 13 | 4 | 10 | 3 | 4 |
| 0.1 | 53 | 108 | 6 | 15 | 6 | 14 | 5 | 6 |
| 0.01 | 64 | 130 | 8 | 19 | 7 | 16 | 8 | 9 |
| 0.001 | 71 | 144 | 12 | 27 | 10 | 20 | 10 | 11 |

♣ 由于采用了精化的调比产生 $\beta_k$ 的策略, 外梯度方法和投影收缩算法的每次迭代几乎只调用 2 次函数值. 就像在第二, 第三讲分析的, 投影收缩算法比外梯度方法收敛快. 外梯度方法的迭代次数和调用 $F(u)$ 的次数都比投影收缩算法的 6–7 倍.

♣ 在投影收缩算法中, 采用 $d_2$ 方向的比采用 $d_1$ 方向的快一些.

♣ 投影梯度法需要的迭代次数与投影收缩算法相当, 但调用 $F(u)$ 的次数比投影收缩算法少一半以上. 求解这类问题, 调用一次 $F(u)$, 相当于调整一次政策. 因此, 在上述算法中, 投影梯度法效率是最高的.

♣ 对变分不等式使用投影梯度法, 要求算子 $F$ 是某个凸函数的梯度.



# 4 An accelerated two-steps P-G method

基于 Nesterov [10] 的思想, 采用 [1] 类似的做法, 可以构造一个只用梯度的快速算法. 这类算法在生成序列 $\{x^k\}$ 的同时, 还生成一个辅助序列 $\{y^k\}$.

**A two-steps projected gradient method**

Take $\beta > 0, x^1 \in R^n$. Set $y^1 = x^1, t_1 = 1$.

Step $k$. ($k \ge 1$) With given $(x^k, y^k)$, let

$$x^{k+1} = P_\Omega[y^k - \beta_k g(y^k)], \tag{4.1a}$$

where the step size $\beta_k$ is chosen to satisfy

$$(y^k - x^{k+1})^T(g(y^k) - g(x^{k+1})) \le \frac{1}{2\beta_k}\|y^k - x^{k+1}\|^2. \tag{4.1b}$$

Set

$$y^{k+1} = x^{k+1} + \Big(\frac{t_k - 1}{t_{k+1}}\Big)(x^{k+1} - x^k), \tag{4.1c}$$

where

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}. \tag{4.1d}$$

The method is called two-steps projected gradient method because each iteration consists of two steps. The $k$-th iteration begins with $(x^k, y^k)$, the first step (4.1a) produces $x^{k+1}$ and the second one (4.1c) updates $y^{k+1}$.

It is assumed that the positive sequence $\{\beta_k\}$ is non-increasing.

We show that the proposed two-steps projected gradient method is convergent with the iteration-complexity $O(1/k^2)$. The proof is similar as those in [1].

**Lemma 4.1** *Let $x^{k+1}$ be given by* (4.1a) *and the step size condition* (4.1b) *be satisfied. Then we have*

$$2\beta_k(f(x)-f(x^{k+1})) \geq \|y^k-x^{k+1}\|^2+2(x^{k+1}-y^k)^T(y^k-x), \;\forall\, x \in \Omega. \quad (4.2)$$

**Proof**. By setting $x^k = y^k$ and $\nu = \frac{1}{2}$ in (2.3a) and (2.3b), we get (4.1a) and (4.1b). Therefore, substituting $x^k = y^k$ and $\nu = \frac{1}{2}$ in (2.5), we get

$$\beta_k(f(x) - f(x^{k+1})) \geq (x - x^{k+1})^T(y^k - x^{k+1}) - \frac{1}{2}\|y^k - x^{k+1}\|^2, \;\forall\, x \in \Omega.$$

The above inequality can be rewritten as (4.2) and the lemma is proved. $\square$

To derive the iteration-complexity of the two-steps projected gradient method, we need to



prove some properties of the corresponding sequence.

**Lemma 4.2** *The sequences $\{x^k\}$ and $\{y^k\}$ generated by the proposed two-steps projected gradient method satisfy*

$$2\beta_k t_k^2 v_k - 2\beta_{k+1} t_{k+1}^2 v_{k+1} \geq \|u^{k+1}\|^2 - \|u^k\|^2, \;\forall k \geq 1, \quad (4.3)$$

*where $v_k := f(x^{k+1}) - f(x^*)$ and $u^k := t_k x^{k+1} - (t_k - 1)x^k - x^*$.*

**Proof**. By using Lemma 4.1 for $k+1$, $x = x^{k+1}$ and $x = x^*$ we get

$$2\beta_{k+1}\big(f(x^{k+1})-f(x^{k+2})\big) \geq \|y^{k+1}-x^{k+2}\|^2+2(x^{k+2}-y^{k+1})^T(y^{k+1}-x^{k+1}),$$

and

$$2\beta_{k+1}\big(f(x^*) - f(x^{k+2})\big) \geq \|y^{k+1} - x^{k+2}\|^2 + 2(x^{k+2} - y^{k+1})^T(y^{k+1} - x^*).$$

Using the definition of $v_k$, we get

$$2\beta_{k+1}(v_k - v_{k+1}) \geq \|y^{k+1} - x^{k+2}\|^2 + 2(x^{k+1} - y^{k+1})^T(y^{k+1} - x^{k+2}), \quad (4.4)$$

and

$$-2\beta_{k+1}v_{k+1} \geq \|y^{k+1} - x^{k+2}\|^2 + 2(x^* - y^{k+1})^T(y^{k+1} - x^{k+2}). \quad (4.5)$$

To get a relation between $v_k$ and $v_{k+1}$, we multiply (4.4) by $(t_{k+1}-1)$ and add it to (4.5):

$$2\beta_{k+1}\big((t_{k+1}-1)v_k - t_{k+1}v_{k+1}\big)$$
$$\geq\ t_{k+1}\|x^{k+2}-y^{k+1}\|^2 + 2(x^{k+2}-y^{k+1})^T\big(t_{k+1}y^{k+1}-(t_{k+1}-1)x^{k+1}-x^*\big).$$

Multiplying the last inequality by $t_{k+1}$ and using

$$t_k^2 = t_{k+1}^2 - t_{k+1} \qquad \big(\text{and thus} \quad t_{k+1} = (1+\sqrt{1+4t_k^2}\,)/2 \quad \text{as in} \quad (4.1\text{d})\big),$$

which yields

$$\begin{array}{rl} & 2\beta_{k+1}\big(t_k^2 v_k - t_{k+1}^2 v_{k+1}\big) \\ \geq & \|t_{k+1}(x^{k+2}-y^{k+1})\|^2 \\ & +2t_{k+1}(x^{k+2}-y^{k+1})^T\big(t_{k+1}y^{k+1}-(t_{k+1}-1)x^{k+1}-x^*\big). \end{array}$$

Applying the relation

$$\|a-b\|^2 + 2(a-b)^T(b-c) = \|a-c\|^2 - \|b-c\|^2$$

to the right-hand side of the last inequality with

$$a := t_{k+1}x^{k+2}, \qquad b := t_{k+1}y^{k+1}, \qquad c := (t_{k+1}-1)x^{k+1} + x^*,$$



and using the fact $2\beta_k t_k^2 v_k \geq 2\beta_{k+1}t_k^2 v_k$ (since $\{\beta_k\}$ is non-increasing), we get

$$\begin{array}{rl} & 2\beta_k t_k^2 v_k - 2\beta_{k+1}t_{k+1}^2 v_{k+1} \\ \geq & \|t_{k+1}x^{k+2}-(t_{k+1}-1)x^{k+1}-x^*\|^2 \\ & -\|t_{k+1}y^{k+1}-(t_{k+1}-1)x^{k+1}-x^*\|^2. \end{array}$$

In order to write the above inequality in the form (4.3) with

$$u^k = t_k x^{k+1} - (t_k - 1)x^k - x^*,$$

we need only to set

$$t_{k+1}y^{k+1} - (t_{k+1}-1)x^{k+1} - x^* = t_k x^{k+1} - (t_k-1)x^k - x^*.$$

From the last equality we obtain

$$y^{k+1} = x^{k+1} + \Big(\frac{t_k - 1}{t_{k+1}}\Big)(x^{k+1}-x^k).$$

This is just the form (4.1c) in the accelerated two-steps version of the projected gradient method. $\square$

To proceed the proof of the main theorem, we need the following Lemma 4.3 and Lemma

4.4, which have also been considered in [1]. We omit their proofs as they are trivial.

**Lemma 4.3** *Let $\{a_k\}$ and $\{b_k\}$ be positive sequences of reals satisfying*

$$a_k - a_{k+1} \geq b_{k+1} - b_k \quad \forall\, k \geq 1.$$

*Then, $a_k \leq a_1 + b_1$ for every $k \geq 1$.*

**Lemma 4.4** *The positive sequence $\{t_k\}$ generated by*

$$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \qquad \text{with} \qquad t_1 = 1$$

*satisfies*

$$t_k \geq \frac{k+1}{2}, \quad \forall\, k \geq 1.$$

Now, we are ready to show that the proposed two-steps projected gradient method is convergent with the rate $O(1/k^2)$.

**Theorem 4.1** *Let $\{x^k\}$ and $\{y^k\}$ be generated by the proposed two-steps projected gradient method. Then, for any $k \geq 1$, we have*

$$f(x^k) - f(x^*) \leq \frac{2\|x^1 - x^*\|^2}{\beta_k k^2}, \quad \forall\, x^* \in \Omega^*. \tag{4.6}$$



**Proof**. Let us define the quantities

$$a_k := 2\beta_k t_k^2 v_k, \qquad b_k := \|u^k\|^2.$$

By using Lemma 4.2 and Lemma 4.3, we obtain

$$2\beta_k t_k^2 v_k \leq a_1 + b_1,$$

which combined with the definition $v_k$ and $t_k \geq (k+1)/2$ (by Lemma 4.4) yields

$$f(x^{k+1}) - f(x^*) = v_k \leq \frac{2(a_1+b_1)}{\beta_k (k+1)^2} \leq \frac{2(a_1+b_1)}{\beta_{k+1}(k+1)^2}. \tag{4.7}$$

Since $t_1 = 1$, and using the definition of $u_k$ given in Lemma 4.2, we have

$$a_1 = 2\beta_1 t_1^2 v_1 = 2\beta_1 v_1 = 2\beta_1\big(f(x^2) - f(x^*)\big), \qquad b_1 = \|u^1\|^2 = \|x^2 - x^*\|^2.$$

Setting $x = x^*$ and $k = 1$ in (4.2), we have

$$\begin{array}{rcl} 2\beta_1(f(x^2) - f(x^*)) & \leq & 2(y^1 - x^*)^T(y^1 - x^2) - \|y^1 - x^2\|^2 \\ & = & \|y^1 - x^*\|^2 - \|x^2 - x^*\|^2. \end{array}$$

Therefore, we have

$$\begin{aligned} a_1 + b_1 &= 2\beta_1(f(x^2) - f(x^*)) + \|x^2 - x^*\|^2 \\ &\le \|y^1 - x^*\|^2 - \|x^2 - x^*\|^2 + \|x^2 - x^*\|^2 \\ &= \|x^1 - x^*\|^2. \end{aligned}$$

Substituting it in (4.7), the assertion is proved. $\Box$

Based on Theorem 2.3, for obtaining an $\varepsilon$-optimal solution (denoted by $\tilde{x}$) in the sense that $f(\tilde{x}) - f(x^*) \le \varepsilon$, the number of iterations required by the proposed two-steps projected gradient method is at most $\lceil C/\sqrt{\varepsilon} - 1\rceil$ where $C = 2\|x^1 - x^*\|^2/\beta$.

需要说明的是, 对这一讲 §5 中的问题, 我们并不提倡用这个附录中的快速方法, 原因是在 (4.1) 的 $k$-次迭代中, 需要至少用到两次梯度的信息, $g(y^k)$ 和 $g(x^{k+1})$. 这里的 $g(\cdot)$ 相当于 §5 中的 $F(\cdot)$. 在实际问题中, $F(\cdot)$ 的获取往往是代价不菲的. 此外, (4.1b) 中要求

$$(y^k - x^{k+1})^T(g(y^k) - g(x^{k+1})) \le \frac{1}{2\beta_k}\|y^k - x^{k+1}\|^2,$$

并要求 $\{\beta_k\}$ 单调不增. 这些条件远不如单步投影梯度法中相应的条件 (2.3b) 宽松. 单步投影梯度法计算实践说明限制 $\{\beta_k\}$ 单调不增会使收敛变慢许多.



# References


[1] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Science, 2 (2009), pp. 183-202.

[2] Y.H. Dai and Y. Yuan, Alternate minimization gradient method. *IMA J. Numerical Analysis* **23** (2003) 377–393.

[3] R. Fletcher, *Practical Methods of Optimization*, Second Edition, John Wiley & Sons, 1987.

[4] B.S. He, *A new method for a class of linear variational inequalities*, Math. Progr., **66**, pp. 137-144, 1994.

[5] B.S. He, *Solving a class of linear projection equations*, Numerische Mathematik, **68**, pp. 71-80, 1994.

[6] B.S He and L-Z Liao, *Improvements of some projection methods for monotone nonlinear variational inequalities*, Journal of Optimization Theory and Applications, **112**, pp. 111-128, 2002

[7] G. M. Korpelevich, The Extragradient Method for Finding Saddle Points and Other Problems, *Ekonomika i Matematchskie Metody* **12** (1976) 747-756.

[8] P. Marcotte, G. Margquis and L. Zubieta, *A Newton-SOR method for spatial price equilibrium*, Transportation Science **26** (1992) 36-47.

[9] A. Nagurney, *An Algorithm for the single commodity spatial price equilibrium problem*, Reg. Sci. and Urban Econ. **16** (1986) 573-588.

[10] Y. E. Nesterov, A method for solving the convex programming problem with convergence rate $O(1/k^2)$, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543-547.

[11] Y. E. Nesterov, On an approach to the construction of optimal methods ofminimization ofsmooth convex functions, Ekonom. Mat. Metody, 24 (1988), pp. 509-517.

[12] P. A. Samuelson, *Spatial price equilibrium and linear programming*, Amer. Ecom. Rev. **42** (1952) 283-303.

# 凸优化和单调变分不等式的收缩算法

## 第九讲: 求解线性约束凸优化
## 基于对偶上升的自适应方法

Self-adaptive dual ascent method for
linearly constrained convex optimization

南京大学数学系　　何炳生
hebma@nju.edu.cn



## 1 Introduction

Let $\mathcal{X}$ be a convex closed set in $\Re^n$. The problem concerned in this note is the constrained convex minimization problem

$$\text{(P)} \qquad \min\{f(x) \,|\, Ax = b,\ x \in \mathcal{X}\}, \tag{1.1}$$

where $f(x) : \Re^n \to \Re$ is a differentiable convex function, $A \in \Re^{m\times n}$ and $b \in \Re^m$. Let $\lambda$ be the Lagrange multiplier to the linear constraints $Ax = b$, the Lagrange function of the problem (1.1) is

$$L(x, \lambda) = f(x) - \lambda^T (Ax - b), \tag{1.2}$$

which is defined on $\mathcal{X} \times \Re^m$. The dual problem of (1.1) is

$$\max \phi(\lambda),$$

where

$$\phi(\lambda) = \inf_{x\in\mathcal{X}} L(x, \lambda).$$

Assuming that strong duality holds, the optimal values of the primal and dual problems are the same. We can recover a primal optimal point $x^*$ from a dual optimal point $\lambda^*$ by solving

$$x^* = \arg\min\{L(x, \lambda^*)|x \in \mathcal{X}\},$$

provided there is only one minimizer of $L(x, \lambda^*)$ (this is the case if, for example, $f$ is strictly convex).

The dual problem can also be interpreted as

$$\text{(D)} \quad \begin{array}{ll} \max_{x,\lambda} & L(x, \lambda) = f(x) - \lambda^T(Ax - b) \\ \text{s. t} & x \in \mathcal{X},\ (x' - x)^T \nabla_x L(x, \lambda) \geq 0,\ \forall x' \in \mathcal{X}. \end{array} \tag{1.3}$$

We denote the solution set of (1.3) by $\Omega^* = \mathcal{X}^* \times \Lambda^*$.

**Dual feasible pair** A pair of $(x, \lambda)$ is dual feasible of (1.3) if and only if

$$x \in \mathcal{X},\ (x' - x)^T(\nabla f(x) - A^T\lambda) \geq 0,\ \forall x' \in \mathcal{X}. \tag{1.4}$$

For any given $\lambda^k \in \Re^m$, we let $x^k$ be defined as in (1.5a). Therefore, $(x^k, \lambda^k)$ is a feasible solution of (1.3). Note that $(x^*, \lambda^*)$ is also dual feasible.



The dual ascent method is the algorithm

$$x^k := \arg\min\{L(x, \lambda^k)|x \in \mathcal{X}\}, \tag{1.5a}$$

and then

$$\lambda^{k+1} = \lambda^k - \alpha_k(Ax^k - b), \tag{1.5b}$$

where $\alpha_k > 0$ is the step size and will be discussed later. The first step (1.5a) is an $x$-minimization step, and the second step (1.5b) is a dual variable update. In some practical applications, the dual variable $\lambda$ can be viewed as a vector of prices, and the $\lambda$-update step is called a price update or price adjustment step. It is called dual ascent since, with appropriate choice of $\alpha$, the dual function increases in each step, i.e.,

$$\phi(\lambda^{k+1}) > \phi(\lambda^k).$$

For any dual feasible pair $(x, \lambda)$, because

$$L(x^*, \lambda^*) = f(x^*) \geq f(x) + (x^* - x)^T \nabla f(x)$$

and

$$(x^* - x)^T(\nabla f(x) - A^T\lambda) \geq 0,$$

it follows that

$$L(x^*, \lambda^*) \geq L(x, \lambda).$$

Thus, $L(x^*, \lambda^*)$ is the maximal value of the dual problem (1.3). By setting

$$v_k = L(x^*, \lambda^*) - L(x^k, \lambda^k), \tag{1.6}$$

the sequence $\{v_k\}$ is non-negative. This paper considers the convergence rate of the non-negative sequence $\{v_k\}$ by the dual ascent method.

**Assumption** Throughout this paper, we assume that the function $f(x)$ is uniformly strict convex. In other words, there is a positive constant $\mu > 0$, such that

$$(x - \tilde{x})^T \big(\nabla f(x) - \nabla f(\tilde{x})\big) \geq \mu \|x - \tilde{x}\|^2, \quad \forall x, \tilde{x}. \tag{1.7}$$

**Lemma 1.1** *Let $(x, \lambda)$ and $(\tilde{x}, \tilde{\lambda})$ be any given dual feasible pairs. Under the assumption* (1.7)*, we have*

$$\|x - \tilde{x}\| \leq \frac{1}{\mu} \|A^T(\lambda - \tilde{\lambda})\|. \tag{1.8}$$

**Proof**. Since $(x, \lambda)$ and $(\tilde{x}, \tilde{\lambda})$ be any given dual feasible pairs, thus we have

$$x \in \mathcal{X}, \quad (\tilde{x} - x)^T \big(\nabla f(x) - A^T\lambda\big) \geq 0,$$



and

$$\tilde{x} \in \mathcal{X}, \quad (x - \tilde{x})^T \big(\nabla f(\tilde{x}) - A^T\tilde{\lambda}\big) \geq 0.$$

Adding the above two inequalities, we obtain

$$(x - \tilde{x})^T A^T (\lambda - \tilde{\lambda}) \geq (x - \tilde{x})^T \big(\nabla f(x) - \nabla f(\tilde{x})\big) \geq \mu \|x - \tilde{x}\|^2,$$

and it follows the assertion (1.8) directly. $\Box$

In other words, under the assumption that $f$ is uniformly strict convex and differentiable, the solution of (1.5a) is a Lipschitz continuous function of $\lambda$.

**Lemma 1.2** *For given $\lambda^k$, let $x^k$ be given by* (1.5a)*. Then for any feasible solution $(x, \lambda)$ of the dual problem* (1.3)*, we have*

$$L(x^k, \lambda^k) - L(x, \lambda) \geq (\lambda - \lambda^k)^T (Ax^k - b). \tag{1.9}$$

**Proof**. First, using the convexity of $f$ we obtain

$$\begin{aligned} & L(x^k, \lambda^k) - L(x, \lambda) \\ = \;& f(x^k) - f(x) + \lambda^T(Ax - b) - (\lambda^k)^T(Ax^k - b) \\ \geq \;& (x^k - x)^T \nabla f(x) + \lambda^T(Ax - b) - (\lambda^k)^T(Ax^k - b). \end{aligned} \tag{1.10}$$

Since $(x, \lambda)$ is a feasible solution of the dual problem and $x^k \in \mathcal{X}$, set $x' = x^k$ in (1.4), we obtain

$$(x^k - x)^T \nabla f(x) \geq (x^k - x)^T A^T \lambda = \lambda^T A(x^k - x).$$

Substituting it in the right hand side of (1.10), we obtain

$$\begin{aligned} & L(x^k, \lambda^k) - L(x, \lambda) \\ & \quad \geq \quad \lambda^T A(x^k - x) + \lambda^T (Ax - b) - (\lambda^k)^T (Ax^k - b) \\ & \quad = \quad (\lambda - \lambda^k)^T (Ax^k - b). \end{aligned}$$

The assertion of this lemma is proved. $\Box$

## 2 Dual ascent method

We assume that $f$ is strictly convex and thus, for any given $\lambda^k$, the $x$-minimization problem (1.5a) has the unique solution $x^k$. Note that this assumption does not hold in many important applications, so dual ascent often cannot be used. As an example, if $f$ is a nonzero affine function of any component of $x$, then the $x$-minimization (1.5a) fails, since



$L(x, \lambda)$ is unbounded below in $x$ for most $\lambda$.

**Dual Ascent Method**(DAM)

Let $(x^0, \lambda^0)$ be a pair of feasible solution of the dual problem (1.3).

For $k = 0, 1, \ldots$, do: Given dual feasible pair $(x^k, \lambda^k)$, set

$$\lambda^{k+1} = \lambda^k - \beta_k (Ax^k - b), \tag{2.1a}$$

and let

$$x^{k+1} = \arg\min \{ L(x, \lambda^{k+1}) \,\big|\, x \in \mathcal{X} \}. \tag{2.1b}$$

The parameter $\beta_k$ is selected such that the dual feasible pairs $(x^k, \lambda^k)$ and $(x^{k+1}, \lambda^{k+1})$ satisfy the condition

$$(\lambda^k - \lambda^{k+1})^T A(x^k - x^{k+1}) \leq \frac{\nu}{\beta_k} \|\lambda^k - \lambda^{k+1}\|^2, \ \nu \in (0, 1). \tag{2.1c}$$

Due to (1.8), by using Armijo-like line search to find a $\beta_k$ to satisfy (2.1c), there is a $\beta_{\min} > 0$ such that

$$\inf_k \{\beta_k\} \geq \beta_{\min}.$$

Note that each pair $(x^k, \lambda^k)$ generated by the dual ascent method is dual feasible.

**Lemma 2.1** *Let* $\{(x^k, \lambda^k)\}$ *be the sequence generated by the dual ascent method* (2.1). *Then we have*

$$L(x^{k+1}, \lambda^{k+1}) - L(x^k, \lambda^k) \geq \frac{1-\nu}{\beta_k}\|\lambda^k - \lambda^{k+1}\|^2. \tag{2.2}$$

**Proof**. Since the sequence $\{(x^k, \lambda^k)\}$ is dual feasible, by setting $(x^k, \lambda^k) = (x^{k+1}, \lambda^{k+1})$ and $(x, \lambda) = (x^k, \lambda^k)$ in (1.9), we obtain

$$L(x^{k+1}, \lambda^{k+1}) - L(x^k, \lambda^k) \geq (\lambda^k - \lambda^{k+1})^T(Ax^{k+1} - b). \tag{2.3}$$

It follows from (2.1c) that

$$\begin{aligned} &(\lambda^k - \lambda^{k+1})^T(Ax^{k+1} - b) \\ &\geq \quad (\lambda^k - \lambda^{k+1})^T(Ax^k - b) - \frac{\nu}{\beta_k}\|\lambda^k - \lambda^{k+1}\|^2. \end{aligned} \tag{2.4}$$

Note that from (2.1a) we have

$$(\lambda^k - \lambda^{k+1})^T(Ax^k - b) = \frac{1}{\beta_k}\|\lambda^k - \lambda^{k+1}\|^2. \tag{2.5}$$

The assertion of this lemma is proved. $\Box$



**Lemma 2.2** *Let* $\{(x^k, \lambda^k)\}$ *be the sequence generated by the dual ascent method* (2.1). *Then for any* $\lambda^* \in \Lambda^*$, *we have*

$$\|\lambda^{k+1}-\lambda^*\|^2 \leq \|\lambda^k-\lambda^*\|^2+\|\lambda^k-\lambda^{k+1}\|^2-2\beta_k\big(L(x^*, \lambda^*)-L(x^k, \lambda^k)\big). \tag{2.6}$$

**Proof**. It follows from (2.1a) that

$$\begin{aligned} &\|\lambda^{k+1} - \lambda^*\|^2 \\ &\quad = \|(\lambda^k - \lambda^*) - (\lambda^k - \lambda^{k+1})\|^2 \\ &\quad = \|\lambda^k - \lambda^*\|^2 + \|\lambda^k - \lambda^{k+1}\|^2 - 2(\lambda^k - \lambda^*)^T\beta_k(Ax^k - b). \end{aligned} \tag{2.7}$$

In addition, because $(x^*, \lambda^*)$ is dual feasible, by setting $(x, \lambda) = (x^*, \lambda^*)$ in (1.9), we obtain

$$(\lambda^k - \lambda^*)^T(Ax^k - b) \geq L(x^*, \lambda^*) - L(x^k, \lambda^k). \tag{2.8}$$

Substituting (2.8) in (2.7), the assertion follows directly. $\Box$

**Lemma 2.3** *Let* $\{(x^k, \lambda^k)\}$ *be the sequence generated by the dual ascent method* (2.1).

*Then for any $\lambda^* \in \Lambda^*$, we have*

$$\begin{aligned} \|\lambda^{k+1} - \lambda^*\|^2 \quad \le \quad & \|\lambda^k - \lambda^*\|^2 - (1-2\nu)\|\lambda^k - \lambda^{k+1}\|^2 \\ & -2\beta_k\big(L(x^*,\lambda^*) - L(x^{k+1},\lambda^{k+1})\big). \end{aligned} \tag{2.9}$$

*Thus the sequence $\{\lambda^k\}$ is Fejèr monotone with respect to $\Lambda^*$ when $\nu \le 1/2$.*

**Proof**. First, it follows from (2.6) that

$$\begin{aligned} &\|\lambda^{k+1} - \lambda^*\|^2 \le \|\lambda^k - \lambda^*\|^2 + \|\lambda^k - \lambda^{k+1}\|^2 \\ &-2\beta_k\big(L(x^*,\lambda^*) - L(x^{k+1},\lambda^{k+1})\big) + 2\beta_k\big(L(x^k,\lambda^k) - L(x^{k+1},\lambda^{k+1})\big). \end{aligned} \tag{2.10}$$

Using (2.2), we get

$$2\beta_k\big(L(x^k,\lambda^k) - L(x^{k+1},\lambda^{k+1})\big) \le -2(1-\nu)\|\lambda^k - \lambda^{k+1}\|^2.$$

Substituting it in (2.10), we get the assertion (2.9). The Lemma is proved. $\Box$

**Theorem 2.1** *Let $\{(x^k,\lambda^k)\}$ be the sequence generated by the dual ascent method* (2.1). *If $\nu \le 1/2$, the sequence $\{\lambda^k\}$ is Fejèr monotone with respect to $\Lambda^*$. Moreover,*

$$L(x^*,\lambda^*) - L(x^k,\lambda^k) \le \frac{1}{2k\beta_{\min}}\|\lambda^0 - \lambda^*\|^2. \tag{2.11}$$



**Proof**. Because $L(x^*,\lambda^*) - L(x^{k+1},\lambda^{k+1}) > 0$ and $\nu \le 1/2$, it follows from (2.9) that

$$\|\lambda^{k+1} - \lambda^*\|^2 < \|\lambda^k - \lambda^*\|^2,$$

whenever $(x^k,\lambda^k) \notin \Omega^*$. Again, from (2.9) we obtain that

$$2\beta_{\min}\big(L(x^*,\lambda^*) - L(x^{l+1},\lambda^{l+1})\big) \le \|\lambda^l - \lambda^*\|^2 - \|\lambda^{l+1} - \lambda^*\|^2.$$

Summing the above inequality over $j = 0,\ldots,k-1$ and using the increasing property of $\{L(x^k,\lambda^k)\}$, the assertion (2.11) follows directly. $\Box$

# 3 Iterations complexity

**Theorem 3.1** *Let $\{(x^k,\lambda^k)\}$ be the sequence generated by the dual ascent method* (2.1) *and $\beta_k \equiv \beta$. Then for any $k \ge 0$ and $(x^*,\lambda^*) \in \mathcal{X}^* \times \Lambda^*$, we have*

$$\begin{aligned} &2k\beta\big(L(x^k,\lambda^k) - L(x^*,\lambda^*)\big) \\ &\quad \ge \sum_{j=0}^{k-1}\Big(2j(1-\nu) + (1-2\nu)\Big)\|\lambda^j - \lambda^{j+1}\|^2 - \|\lambda^0 - \lambda^*\|^2. \end{aligned} \tag{3.1}$$

**Proof**. It follows from (2.9) that, for all $j \ge 0$, we have

$$
\begin{aligned}
&2\beta(L(x^{j+1},\lambda^{j+1}) - L(x^*,\lambda^*))\\
&\quad\ge \quad \|\lambda^{j+1}-\lambda^*\|^2 - \|\lambda^j-\lambda^*\|^2 + (1-2\nu)\|\lambda^j-\lambda^{j+1}\|^2.
\end{aligned}
$$

Using the fact $L(x^j,\lambda^j) - L(x^*,\lambda^*) < 0$, summing the above inequality over $j = 0,\ldots,k-1$, we obtain

$$
\begin{aligned}
&2\beta\Big(\sum_{j=0}^{k-1} L(x^{j+1},\lambda^{j+1}) - kL(x^*,\lambda^*)\Big)\\
&\quad\ge \quad \|\lambda^k-\lambda^*\|^2 - \|\lambda^0-\lambda^*\|^2 + (1-2\nu)\sum_{j=0}^{k-1}\|\lambda^j-\lambda^{j+1}\|^2. \qquad (3.2)
\end{aligned}
$$

By using Lemma 2.1 for $k = j$, we get

$$\beta(L(x^{j+1},\lambda^{j+1}) - L(x^j,\lambda^j)) \ge (1-\nu)\|\lambda^j-\lambda^{j+1}\|^2.$$



Multiplying the last inequality by $2j$ and summing over $j = 0,\ldots,k-1$, it follows that

$$
\begin{aligned}
&2\beta\sum_{j=0}^{k-1}\Big((j+1)L(x^{j+1},\lambda^{j+1}) - L(x^{j+1},\lambda^{j+1}) - jL(x^j,\lambda^j)\Big)\\
&\quad\ge \quad \sum_{j=0}^{k-1} 2j(1-\nu)\|\lambda^j-\lambda^{j+1}\|^2,
\end{aligned}
$$

which simplifies to

$$2\beta\Big(kL(x^k,\lambda^k) - \sum_{j=0}^{k-1} L(x^{j+1},\lambda^{j+1})\Big) \ge \sum_{j=0}^{k-1} 2j(1-\nu)\|\lambda^j-\lambda^{j+1}\|^2. \qquad (3.3)$$

Adding (3.2) and (3.3), we get

$$
\begin{aligned}
&2k\beta\big(L(x^k,\lambda^k) - L(x^*,\lambda^*)\big)\\
&\quad\ge \quad \sum_{j=0}^{k-1}\Big(2j(1-\nu) + (1-2\nu)\Big)\|\lambda^j-\lambda^{j+1}\|^2 - \|\lambda^0-\lambda^*\|^2.
\end{aligned}
$$

The proof is complete. $\square$

♣ In fact, for any $\nu \in (0,1)$, the iteration-complexity of the single projected gradient method is $O(1/k)$. For example, if $\nu = 0.9$, then

$$(1-2\nu) + 2j(1-\nu) \geq 0, \ \forall\, j \geq 4.$$

In this case, it follows from (3.1) that

$$\begin{aligned} & 2k\beta\big(L(x^*,\lambda^*) - L(x^k,\lambda^k)\big) \\ & \quad \leq \quad \|\lambda^0 - \lambda^*\|^2 + \textstyle\sum_{l=0}^{3}\Big((2\nu-1) + 2l(\nu-1)\Big)\|\lambda^j - \lambda^{j+1}\|^2. \end{aligned} \tag{3.4}$$

Since $\nu \leq 0.9$, we have

$$L(x^*,\lambda^*) - L(x^k,\lambda^k) \leq \frac{1}{2k\beta}\Big(\|\lambda^0 - \lambda^*\|^2 + \tfrac{4}{5}\textstyle\sum_{l=0}^{3}\|\lambda^j - \lambda^{j+1}\|^2\Big). \tag{3.5}$$

**Practical Condition** In practical computation, instead of the condition (2.1c), we use

$$\|\beta_k A(x^k - x^{k+1})\| \leq \nu\|\lambda^k - \lambda^{k+1}\|, \quad \nu \in (0,1) \tag{3.6}$$

as the acceptance condition. It is clear that above condition is stronger than the condition (2.1c). We use the following self-adaptive dual ascent method:



Self-adaptive dual ascent method:

Step 0. Set $\beta_0 = 1$, $\nu \in (0,1)$, $\lambda^0 \in \Re^m$, $x^0 = \arg\min\{L(x,\lambda^0) | x \in \mathcal{X}\}$.

For $k = 0, 1, \ldots$, if the stopping criterium is not satisfied, do:

Step 1. $\tilde{\lambda}^k = \lambda^k - \beta_k(Ax^k - b)$, $\quad \tilde{x}^k = \arg\min\{L(x,\tilde{\lambda}^k) \,|\, x \in \mathcal{X}\}$,

$r_k := \|\beta_k A(x^k - \tilde{x}^k)\| / \|\lambda^k - \tilde{\lambda}^k\|$,

**while** $\ r_k > \nu$

$\beta_k := \frac{2}{3}\beta_k * \min\{1, \frac{1}{r_k}\}$,

$\tilde{\lambda}^k = \lambda^k - \beta_k(Ax^k - b)$, $\quad \tilde{x}^k = \arg\min\{L(x,\tilde{\lambda}^k) \,|\, x \in \mathcal{X}\}$,

$r_k := \|\beta_k A(x^k - \tilde{x}^k)\| / \|\lambda^k - \tilde{\lambda}^k\|$,

**end(while)**

$\lambda^{k+1} = \tilde{\lambda}^k$,

**If** $\ r_k \leq \mu\ $ then $\ \beta_k := \beta_k * 1.5$, **end(if)**

Step 2. $\beta_{k+1} = \beta_k$ and $k = k+1$, go to Step 1.

采用上述程序但略去 **If** $r_k \leq \mu$ then $\beta_k := \beta_k * 1.5$ **end(if)** 的做法, 将大大增加迭代步数, 有时甚至导致计算失败.

# 4 Applications of the self-adaptive dual-ascent method

在统计学中, 一个对角元均为 1 的对称半正定矩阵称为相关性矩阵 (Correlation Matrix). 对给定的对称矩阵 $C$, 求 $F$-模下与 $C$ 距离最近的相关性矩阵, 其数学表达式是

$$\min\{\frac{1}{2}\|X-C\|_F^2 \mid \operatorname{diag}(X)=e,\ X\in S_+^n\}, \tag{4.1}$$

其中 $e$ 表示每个分量都为 $1$ 的 $n$-维向量, $S_+^n$ 表示 $n\times n$ 正半定锥的集合. 问题 (4.1) 是形如 (1.1) 的等式约束凸优化问题, 其中 $\|A^TA\|=1$.

我们用 $z\in\Re^n$ 作为等式约束 $\operatorname{diag}(X)=e$ 的 Lagrange 乘子.

**用第四讲的 PPA 算法求解问题** (4.1), 具体做法可见第四讲的 $\S\ 4.1$.

对给定的 $z^k$, 产生 $\tilde{X}^k$ 的方法是:

$$\min\{\frac{1}{2}\|X-C\|_F^2-(z^k)^T(\text{diag}(X)-e)|X\in S_+^n\}. \tag{4.2}$$



**子问题** (4.2) **求解的具体做法:** 化为等价问题

$$\min\{\frac{1}{2}\|X-(C+\operatorname{diag}(z^k))\|_F^2|X\in S_+^n\}.$$

因此我们只要考虑如何求解

$$\tilde{X}^k=\text{Argmin}\{\frac{1}{2}\|X-A\|_F^2\mid X\in S_+^n\}. \tag{4.3}$$

问题 (4.3) 的解法在第四讲里已经作了介绍.

采用 **Dual Ascent Method**, 每步迭代中最大的花费是要对给定的 $\lambda^k$, 生成一个 Dual feasible pair $(x^k,\lambda^k)$, 其中

$$x^k=\text{Argmin}\{L(x,\lambda^k)|x\in\mathcal{X}\}.$$

这个子问题的形式就是 (4.2), 我们只是将它的解及成 $X^k$.

我们对不同的方法进行对比计算. 在 Matlab 程序中, 对称矩阵特征值分解, 都使用 mexeig 子程序. 试验结果表明, 对这一类问题, **Dual Ascent Method** 比第四讲的 **Customized PPA** 还要快一倍左右.

## Code 5.A. Matlab code for Creating the test examples

```
%%%%    DEMO        Comparison the two different methods %%%%%%%%%%%%
%%%  min { (1/2)|X-C|^2 | X is positive semi-definite, X_{jj}=1 }  %%%
clear; close all; clc;                                          %(1)
n = 500;            tol=1e-6;                                   %(2)
%% Generating the given matrix C                                %(3)
rand('state',0);          randn('state',0);                     %(4)
C=rand(n,n);               C=(C'+C) - ones(n,n) + eye(n);       %(5)
%% C is symmetric, C_{ij} in (-1,1) for i\ne j,  C_{jj} in (0,2)  %(6)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%     %(7)
%% Run  Extende PPA with mexeig %%                              %(8)
   r = 2.0;   s = 1.01/r;    gamma = 1.5;     %% Given Parameter  %(9)
   PPA_G(n,C,r,s,tol,gamma)                                    %(10)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    %(11)
 %% Run Dual-Ascent Method                                     %(12)
   beta=1.0;                                  %% Given Parameter %(13)
   Dual_A(n,C,beta,tol)                                        %(14)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

生成的对称矩阵 $C$, 对角元在 $(0,2)$ 之间, 非对角元在 $(-1,1)$ 之间.



## Code 5.1 Matlab Code of the Extended PPA

```
%%%     Extended PPA for calibrating correlation matrix          %(1)
function PPA_G(n,C,r,s,tol,gamma)                                %(2)
X = eye(n);      y = zeros(n,1);    tic;  %% The initial iterate   %(3)
stopc=1;       k=0;                                              %(4)
while (stopc>tol && k<=100)          %% Beginning of an Iteration  %(5)
    if mod(k,1)==0; fprintf('k=%3d  epsm=%9.3e\n',k,stopc);  end;  %(6)
    X0 = X;      y0 = y;            k=k+1;                       %(7)
    yt = y0 - (diag(X0)-ones(n,1))/s;                            %(8)
    A =(X0*r + C + diag(yt*2-y0))/(1+r);                         %(9)
     [V,D] = mexeig(A);   D = max(0,D);  XT = (V*D)*V'; %% mexeig %(10)
     EX = X0-XT;           EY = y0-yt;                          %(11)
     ex = max(max(abs(EX)));  ey = max(abs(EY));                %(12)
     stopc= max(ex,ey);                                         %(13)
    X = X0-EX*gamma;         y = y0-EY*gamma;                   %(14)
end                                     %% End of an Iteration %(15)
toc                                                             %(16)
TB = max(abs(diag(X-eye(n))));                                  %(17)
 fprintf('k=%3d epsm=%9.3e  TB=%8.5f\n\n',k,stopc,TB);          %(18)
```

## Code 5.2 Matlab Code of Dual Ascent Method

```
%%%   Dual-Ascent-Method, Dual variable z                         %(1)
function Dual_A(n,C,beta,tol)                                     %(2)
 z=zeros(n,1);       tic;            %%  The initial iterate      %(3)
 A= C + diag(z);  [V,D]=mexeig(A);  D=max(0,D);    X=(V*D)*V';    %(4)
  r=1;   k=0;      l=0;      stopc=1;                             %(5)
while (stopc>tol && k<=60)        %% Beginning of an Iteration    %(6)
     gz= diag(X)-ones(n,1);       stopc=max(abs(gz));             %(7)
        k= k+1;        l=l+1;                                     %(8)
        dz=gz*beta;        zt=z-dz;        A= C + diag(zt);       %(9)
        [V,D]=mexeig(A);  D=max(0,D);      XT=(V*D)*V';          %(10)
        df=(diag(X)-diag(XT))*beta;        r=norm(df)/norm(dz);  %(11)
        while r>0.9                                              %(12)
          beta=0.8*beta;    l=l+1;                               %(13)
          dz=gz*beta;     zt=z-dz;        A= C + diag(zt);       %(14)
          [V,D]=mexeig(A);  D=max(0,D);    XT=(V*D)*V';          %(15)
          df=(diag(X)-diag(XT))*beta;      r=norm(df)/norm(dz);  %(16)
        end;                                                     %(17)
        z = zt;    X=XT;     if r <0.6    beta=beta*1.5;   end;  %(18)
 end;                                %% End of an Iteration      %(19)
toc;     fprintf('  k=%4d   epsm=%9.3e   l=%4d \n',k,stopc,l);   %(20)
```



### 矩阵校正问题 (4.1)　　精度要求 $\varepsilon = 10^{-4}$

| $n \times n$ Matrix | Extended PPA | | Dual-Ascent Method | | |
|---|---|---|---|---|---|
| $n =$ | No. It | CPU Sec. | No. It | No. of Solving Sub-Prob (2.1b) | CPU Sec. |
| 100 | 18 | 0.11 | 11 | 11 | 0.07 |
| 200 | 21 | 0.36 | 12 | 12 | 0.21 |
| 500 | 22 | 3.41 | 12 | 12 | 1.80 |
| 800 | 24 | 12.65 | 13 | 13 | 7.05 |
| 1000 | 25 | 24.75 | 13 | 13 | 12.51 |
| 1500 | 30 | 93.74 | 13 | 13 | 42.42 |
| 2000 | 34 | 241.25 | 14 | 14 | 103.85 |

The dual ascent method converges much faster than the extended PPA.

$$\frac{\text{CPU. Time of the dual ascent method}}{\text{CPU. Time of the extended PPA}} \le \begin{cases} 55\% & n < 1000 \\ 45\% & n \ge 1000 \end{cases}$$

矩阵校正问题 (4.1)　　精度要求 $\varepsilon = 10^{-6}$

| $n \times n$ Matrix | Extended PPA | | Dual-Ascent Method | | |
|---|---|---|---|---|---|
| $n =$ | No. It | CPU Sec. | No. It | No. of Solving Sub-Prob (2.1b) | CPU Sec. |
| 100 | 26 | 0.16 | 14 | 14 | 0.09 |
| 200 | 29 | 0.50 | 17 | 17 | 0.29 |
| 500 | 32 | 4.96 | 17 | 17 | 2.54 |
| 800 | 35 | 18.45 | 19 | 19 | 10.29 |
| 1000 | 36 | 35.64 | 17 | 17 | 18.75 |
| 1500 | 44 | 137.50 | 18 | 18 | 58.74 |
| 2000 | 50 | 354.78 | 20 | 20 | 148.36 |

The dual ascent method converges much faster than the extended PPA.

$$\frac{\text{CPU. Time of the dual ascent method}}{\text{CPU. Time of the extended PPA}} \le \begin{cases} 55\% & n < 1000 \\ 45\% & n \ge 1000 \end{cases}$$



# 5 An accelerated two-steps dual ascent method

According to the basic idea of Nesterov [6], we can construct the accelerated two-steps dual ascent method. Besides $\{\lambda^k\}$, it generates an auxiliary sequence $\{\eta^k\}$.

**A two-steps dual ascent method**

Step 0. Take $\beta > 0$, $\lambda^1 \in R^n$. Set $\eta^1 = \lambda^1$, $t_1 = 1$.

Step $k$. ($k \ge 1$) With given $(\lambda^k, \eta^k)$, produce the dual feasible pair $(x_\eta^k, \eta^k)$ and let

$$\lambda^{k+1} = \eta^k - \beta_k(Ax_\eta^k - b), \tag{5.1a}$$

then generate the new dual feasible pair $(x^{k+1}, \lambda^{k+1})$. The step size $\beta_k$ should ensure the two dual feasible pairs, $(x_\eta^k, \eta^k)$ and $(x^{k+1}, \lambda^{k+1})$, to satisfy

$$(\eta^k - \lambda^{k+1})^T A(x_\eta^k - x^{k+1}) \le \frac{1}{2\beta_k}\|\eta^k - \lambda^{k+1}\|^2. \tag{5.1b}$$

Set

$$\eta^{k+1} = \lambda^{k+1} + \Big(\frac{t_k - 1}{t_{k+1}}\Big)(\lambda^{k+1} - \lambda^k), \tag{5.1c}$$

where

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}. \tag{5.1d}$$

The method is called two-steps dual ascent method because each iteration consists of two steps. The $k$-th iteration begins with $(\lambda^k, \eta^k)$, the first step (5.1a) produces $\lambda^{k+1}$ and the second one (5.1c) updates $\eta^{k+1}$. In each iteration, it needs at least to produce two dual feasible pairs, namely

$$(x_\eta^k, \eta^k) \quad \text{and} \quad (x^{k+1}, \lambda^{k+1}).$$

It is assumed that the positive sequence $\{\beta_k\}$ is non- increasing. We show that the proposed two-steps dual ascent method is convergent with the iteration-complexity $O(1/k^2)$. The proof is similar as those in [1].

**Lemma 5.1** *Let* $\lambda^{k+1}$ *be given by* (5.1a) *and the step size condition* (5.1b) *be satisfied. Then we have*

$$\begin{array}{rl} & 2\beta_k(L(x^{k+1}, \lambda^{k+1}) - L(x, \lambda)) \\ \geq & \|\eta^k - \lambda^{k+1}\|^2 + 2(\lambda^{k+1} - \eta^k)^T(\eta^k - \lambda), \ \forall\, \lambda \in \Re^m. \end{array} \tag{5.2}$$

**Proof**. By using (1.9), for any feasible solution $(x, \lambda)$ of the dual problem (1.3), we get

$$L(x_\eta^k, \eta^k) - L(x, \lambda) \geq (\lambda - \eta^k)^T(Ax_\eta^k - b).$$



Due to (5.1a), we have

$$\begin{array}{rcl} (\lambda - \eta^k)^T(Ax_\eta^k - b) & = & \{(\lambda - \lambda^{k+1}) + (\lambda^{k+1} - \eta^k)\}^T(Ax_\eta^k - b) \\ & = & \dfrac{1}{\beta_k}(\lambda - \lambda^{k+1})(\eta^k - \lambda^{k+1}) - \dfrac{1}{\beta_k}\|\eta^k - \lambda^{k+1}\|^2, \end{array}$$

and consequently

$$L(x_\eta^k, \eta^k) - L(x, \lambda) \geq \frac{1}{\beta_k}(\lambda - \lambda^{k+1})(\eta^k - \lambda^{k+1}) - \frac{1}{\beta_k}\|\eta^k - \lambda^{k+1}\|^2. \tag{5.3}$$

Again, by setting $k := k+1$ and $(x, \lambda) = (x_\eta^k, \eta^k)$ in (1.9), we obtain

$$\begin{array}{rl} & L(x^{k+1}, \lambda^{k+1}) - L(x_\eta^k, \eta^k) \\ \geq & (\eta^k - \lambda^{k+1})^T(Ax^{k+1} - b) \\ = & (\eta^k - \lambda^{k+1})^T\{(Ax_\eta^k - b) - A(x_\eta^k - x^{k+1})\} \\ = & \dfrac{1}{\beta_k}\|\eta^k - \lambda^{k+1}\|^2 - (\eta^k - \lambda^{k+1})^T A(x_\eta^k - x^{k+1}) \\ \geq & \dfrac{1}{2\beta_k}\|\eta^k - \lambda^{k+1}\|^2. \end{array} \tag{5.4}$$

Adding (5.3) and (5.4),

$$
\begin{aligned}
& L(x^{k+1}, \lambda^{k+1}) - L(x, \lambda) \\
\geq \;& \frac{1}{\beta_k}(\lambda - \lambda^{k+1})(\eta^k - \lambda^{k+1}) - \frac{1}{2\beta_k}\|\eta^k - \lambda^{k+1}\|^2 \\
= \;& \frac{1}{\beta_k}(\lambda - \eta^k)^T(\eta^k - \lambda^{k+1}) + \frac{1}{2\beta_k}\|\eta^k - \lambda^{k+1}\|^2. \qquad (5.5)
\end{aligned}
$$

The above inequality can be rewritten as (5.2) and the lemma is proved. $\square$

To derive the iteration-complexity of the two-steps projected gradient method, we need to prove some properties of the corresponding sequence.

**Lemma 5.2** *The sequences $\{\lambda^k\}$ and $\{\eta^k\}$ generated by the proposed two-steps dual ascent method satisfy*

$$
2\beta_k t_k^2 v_k - 2\beta_{k+1} t_{k+1}^2 v_{k+1} \geq \|u^{k+1}\|^2 - \|u^k\|^2, \; \forall k \geq 1, \qquad (5.6)
$$

*where $v_k := L(x^*, \lambda^*) - L(x^{k+1}, \lambda^{k+1})$ and $u^k := t_k \lambda^{k+1} - (t_k - 1)\lambda^k - \lambda^*$.*



**Proof**. By using Lemma 5.1 for $k+1$, $x = \lambda^{k+1}$ and $x = \lambda^*$ we get

$$
\begin{aligned}
& 2\beta_{k+1}\big(L(x^{k+2}, \lambda^{k+2}) - L(x^{k+1}, \lambda^{k+1})\big) \\
\geq \;& \|\eta^{k+1} - \lambda^{k+2}\|^2 + 2(\lambda^{k+2} - \eta^{k+1})^T(\eta^{k+1} - \lambda^{k+1}),
\end{aligned}
$$

and

$$
\begin{aligned}
& 2\beta_{k+1}\big(L(x^{k+2}, \lambda^{k+2}) - L(x^*, \lambda^*)\big) \\
\geq \;& \|\eta^{k+1} - \lambda^{k+2}\|^2 + 2(\lambda^{k+2} - \eta^{k+1})^T(\eta^{k+1} - \lambda^*).
\end{aligned}
$$

Using the definition of $v_k$, we get

$$
2\beta_{k+1}(v_k - v_{k+1}) \geq \|\eta^{k+1} - \lambda^{k+2}\|^2 + 2(\lambda^{k+1} - \eta^{k+1})^T(\eta^{k+1} - \lambda^{k+2}), \qquad (5.7)
$$

and

$$
-2\beta_{k+1} v_{k+1} \geq \|\eta^{k+1} - \lambda^{k+2}\|^2 + 2(\lambda^* - \eta^{k+1})^T(\eta^{k+1} - \lambda^{k+2}). \qquad (5.8)
$$

To get a relation between $v_k$ and $v_{k+1}$, we multiply (5.7) by $(t_{k+1}-1)$ and add it to (5.8):

$$\begin{aligned}
&2\beta_{k+1}\big((t_{k+1}-1)v_k - t_{k+1}v_{k+1}\big)\\
&\quad\geq\quad t_{k+1}\|\lambda^{k+2}-\eta^{k+1}\|^2\\
&\qquad\quad +2(\lambda^{k+2}-\eta^{k+1})^T\big(t_{k+1}\eta^{k+1}-(t_{k+1}-1)\lambda^{k+1}-\lambda^*\big).
\end{aligned}$$

Multiplying the last inequality by $t_{k+1}$ and using

$$t_k^2 = t_{k+1}^2 - t_{k+1} \qquad \big(\text{and thus} \quad t_{k+1} = (1+\sqrt{1+4t_k^2}\,)/2 \quad \text{as in} \quad (5.1\text{d})\big),$$

which yields

$$\begin{aligned}
&2\beta_{k+1}\big(t_k^2 v_k - t_{k+1}^2 v_{k+1}\big)\\
&\quad\geq\quad \|t_{k+1}(\lambda^{k+2}-\eta^{k+1})\|^2\\
&\qquad\quad +2t_{k+1}(\lambda^{k+2}-\eta^{k+1})^T\big(t_{k+1}\eta^{k+1}-(t_{k+1}-1)\lambda^{k+1}-\lambda^*\big).
\end{aligned}$$

Applying the relation

$$\|a-b\|^2 + 2(a-b)^T(b-c) = \|a-c\|^2 - \|b-c\|^2$$



to the right-hand side of the last inequality with

$$a := t_{k+1}\lambda^{k+2}, \qquad b := t_{k+1}\eta^{k+1}, \qquad c := (t_{k+1}-1)\lambda^{k+1} + \lambda^*,$$

and using the fact $2\beta_k t_k^2 v_k \geq 2\beta_{k+1} t_k^2 v_k$ (since $\{\beta_k\}$ is non-increasing), we get

$$\begin{aligned}
&2\beta_k t_k^2 v_k - 2\beta_{k+1} t_{k+1}^2 v_{k+1}\\
&\quad\geq\quad \|t_{k+1}\lambda^{k+2} - (t_{k+1}-1)\lambda^{k+1} - \lambda^*\|^2\\
&\qquad\quad -\|t_{k+1}\eta^{k+1} - (t_{k+1}-1)\lambda^{k+1} - \lambda^*\|^2.
\end{aligned}$$

In order to write the above inequality in the form (5.6) with

$$u^k = t_k\lambda^{k+1} - (t_k-1)\lambda^k - \lambda^*,$$

we need only to set

$$t_{k+1}\eta^{k+1} - (t_{k+1}-1)\lambda^{k+1} - \lambda^* = t_k\lambda^{k+1} - (t_k-1)\lambda^k - \lambda^*.$$

From the last equality we obtain

$$\eta^{k+1} = \lambda^{k+1} + \Big(\frac{t_k-1}{t_{k+1}}\Big)(\lambda^{k+1}-\lambda^k).$$

This is just the form (5.1c) in the accelerated two-steps version of the dual ascent method and the lemma is proved. □

To proceed the proof of the main theorem, we need the following Lemma 5.3 and Lemma 5.4, which have also been considered in [1]. We omit their proofs as they are trivial.

**Lemma 5.3** *Let $\{a_k\}$ and $\{b_k\}$ be positive sequences of reals satisfying*

$$a_k - a_{k+1} \geq b_{k+1} - b_k \quad \forall\, k \geq 1.$$

*Then, $a_k \leq a_1 + b_1$ for every $k \geq 1$.*

**Lemma 5.4** *The positive sequence $\{t_k\}$ generated by*

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \qquad \text{with} \qquad t_1 = 1$$

*satisfies*

$$t_k \geq \frac{k+1}{2}, \quad \forall\, k \geq 1.$$

Now, we are ready to show that the proposed two-steps projected gradient method is convergent with the rate $O(1/k^2)$.

**Theorem 5.1** *Let $\{\lambda^k\}$ and $\{\eta^k\}$ be generated by the proposed two-steps dual ascent*



*method . Then, for any $k \geq 1$, we have*

$$L(x^*, \lambda^*) - L(x^k, \lambda^k) \leq \frac{2\|\lambda^1 - \lambda^*\|^2}{\beta_k k^2}, \quad \forall\, \lambda^* \in \Omega^*. \tag{5.9}$$

**Proof**. Let us define the quantities

$$a_k := 2\beta_k t_k^2 v_k, \qquad b_k := \|u^k\|^2.$$

By using Lemma 5.2 and Lemma 5.3, we obtain

$$2\beta_k t_k^2 v_k \leq a_1 + b_1,$$

which combined with the definition $v_k$ and $t_k \geq (k+1)/2$ (by Lemma 5.4) yields

$$L(x^*, \lambda^*) - L(x^{k+1}, \lambda^{k+1}) = v_k \leq \frac{2(a_1 + b_1)}{\beta_k (k+1)^2} \leq \frac{2(a_1 + b_1)}{\beta_{k+1} (k+1)^2}. \tag{5.10}$$

Since $t_1 = 1$, and using the definition of $u_k$ given in Lemma 5.2, we have

$$a_1 = 2\beta_1 t_1^2 v_1 = 2\beta_1 v_1 = 2\beta_1 \big(L(x^*, \lambda^*) - L(x^2, \lambda^2)\big),$$

and

$$b_1 = \|u^1\|^2 = \|\lambda^2 - \lambda^*\|^2.$$

Setting $\lambda = \lambda^*$ and $k = 1$ in (5.2), we have

$$\begin{aligned} 2\beta_1(L(x^*, \lambda^*) - L(x^2, \lambda^2)) &\leq 2(\eta^1 - \lambda^*)^T(\eta^1 - \lambda^2) - \|\eta^1 - \lambda^2\|^2 \\ &= \|\eta^1 - \lambda^*\|^2 - \|\lambda^2 - \lambda^*\|^2. \end{aligned}$$

Therefore, we have

$$\begin{aligned} a_1 + b_1 &= 2\beta_1(L(x^*, \lambda^*) - L(x^2, \lambda^2)) + \|\lambda^2 - \lambda^*\|^2 \\ &\leq \|\eta^1 - \lambda^*\|^2 - \|\lambda^2 - \lambda^*\|^2 + \|\lambda^2 - \lambda^*\|^2 \\ &= \|\lambda^1 - \lambda^*\|^2. \end{aligned}$$

Substituting it in (5.10), the assertion is proved. $\Box$

Based on Theorem 5.1, for obtaining an $\varepsilon$-optimal dual solution (denoted by $\lambda$) in the sense that $L(x^*, \lambda^*) - L(x, \lambda) \leq \varepsilon$, the number of iterations required by the proposed two- steps dual ascent method is at most $\lceil C/\sqrt{\varepsilon} - 1 \rceil$ where $C = 2\|\lambda^1 - \lambda^*\|^2/\beta$.



# 6 Conclusion remarks

According to my limited numerical experiences, it is very important to adjust the parameter $\beta$ in the self-adaptive dual ascent method in Section 3. A suitable small $\beta$ in (2.1a) will ensure the condition (2.1c) and the convergence. However, if

$$r_k = \frac{\|\beta_k A(x^k - \tilde{x}^k)\|}{\|\lambda^k - \tilde{\lambda}^k\|} \leq \mu, \quad (\text{say } \mu = 0.4)$$

the parameter $\beta$ should be to enlarge for the trial in the next iteration.

Notice that, in the convergence rate proof of the accelerated two-steps dual ascent method, it is assumed that the nonnegative positive sequence $\{\beta_k\}$ is non-increasing. This “non-increasing” assumption will destroy the convergence behaviours in the practical computation.

# References


[1] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Science, 2 (2009), pp. 183-202.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Foundations and Trends in Machine Learning, 3 (2010), pp. 1-122.

[3] B. S. He, A class of projection and contraction methods for monotone variational inequalities, *Applied Mathematics and optimization*, 35: pp. 69–76, 1997.

[4] B. S. He and L. Z. Liao, Improvements of some projection methods for monotone nonlinear variational inequalities. *Journal of Optimization Theory and Applications* **112** (2002) 111-128.

[5] A. S. Nemirovsky and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons, New York, 1983.

[6] Y. E. Nesterov, A method for solving the convex programming problem with convergence rate $O(1/k^2)$, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543-547.

[7] Y. E. Nesterov, Gradient methods for minimizing composite objective function, CORE report 2007; available at http://www.ecore.be/DPs/dp-1191313936.pdf.

[8] M.J.D. Powell, The Lagrange method and SAO with bounds on the dual variables, Dec. 2011, http://www.optimization-online.org/DB_FILE/2011/12/3295.pdf

[9] R.T. Rockafellar, Monotone operators and the proximal point algorithm, *SIAM J. Control Optimization*, 14: pp. 877-898, 1976.

# 凸优化和单调变分不等式的收缩算法

## 第十讲: 线性约束单调变分不等式的投影收缩算法

Projection and Contraction Method for linearly constrained monotone variational inequalities

南京大学数学系　　何炳生

hebma@nju.edu.cn

The context of this lecture is based on the publications [2]



## 1 Introduction

Let $A \in \Re^{n\times m}$ and $b \in \Re^m$, $\mathcal{X} \subset \Re^n$ be a nonempty closed convex set and $f$ be a continuous and monotone mapping from $\Re^n$ into itself. In this paper, we focus on the following constrained variational inequality problem

$$\text{Find} \quad x^* \in S \quad \text{such that} \qquad (x - x^*)^T f(x^*) \geq 0, \quad \forall x \in S, \tag{1.1}$$

where

$$S = \{x \in \Re^n \mid A^T x = b, \; x \in \mathcal{X}\} \tag{1.2}$$

or

$$S = \{x \in \Re^n \mid A^T x \leq b, \; x \in \mathcal{X}\}. \tag{1.3}$$

In transportation equilibrium problem, the vector $x$ is the path-flow. Usually, $A$ denotes the path-link incidence matrix and thus the link flow vector is $A^T x$. The vector $b$ denotes the link capacity. Therefore, we use (1.3) to denote the constraint set in the problem (1.1).

The method in this lecture is based on the paper [2]. We assume that $f$ does not have any explicit form and only its value can be evaluated for given variable. Moreover, we assume that the projection on $\mathcal{X}$ is simple to carry out.

By introducing the Lagrangian multiplier $y \in \mathcal{Y} = \mathcal{R}^m$ and $y \in \mathcal{Y} = \Re^m_+$ to the linear constraints $A^T x - b = 0$ and $A^T x - b \le 0$, respectively, problem (1.1) can be translated to an enlarged variational inequality VI$(\Omega, F)$:

$$\text{Find} \quad u^* \in \Omega \quad \text{such that} \quad (u - u^*)^T F(u^*) \ge 0, \qquad \forall u \in \Omega, \tag{1.4}$$

where

$$u = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad F(u) = \begin{pmatrix} f(x) + Ay \\ -A^T x + b \end{pmatrix} \qquad \text{and} \qquad \Omega = \mathcal{X} \times \mathcal{Y}. \tag{1.5}$$

Since the projection on $\Omega$ is trivial, problem VI$(\Omega, F)$ can be solved by some projection type methods [4, 6] which only use the function value of $F$ in the solution process.

Let $G =$ diag$(I_n, \nu I_m)$ be a positive definite diagonal matrix. For given $u^k \in \Omega$ and $\beta_k > 0$, the existing proximal point algorithm (abbreviated as PPA) under $G$-norm for VI$(\Omega, F)$ generate a new iterate $u^{k+1} \in \Omega$ which is the unique solution of the following problem:

$$u \in \Omega, \quad (u' - u)^T \{G(u - u^k) + \beta_k F(u)\} \ge 0, \quad \forall u' \in \Omega. \tag{1.6}$$

We use the notations $P_\Omega(\cdot)$ and $P_{\Omega,G}(\cdot)$ to denote the projection under $l_2$ norm and



$G$-norm, respectively. Since $G$ is a block scalar matrix and $\Omega = \mathcal{X} \times \mathcal{Y}$, For given $v$,

$$\text{Argmin}\{\|v - u\|_G \mid u \in \Omega\} = \text{Argmin}\{\|v - u\| \mid u \in \Omega\}$$

Thus, an equivalent expression of (1.6) is the following projection equation

$$u = P_\Omega\{u^k - \beta_k G^{-1} F(u)\}. \tag{1.7}$$

The sequence $\{u^k\}$ generated by PPA satisfies

$$\|u^{k+1} - u^*\|_G^2 \le \|u^k - u^*\|_G^2 - \|u^k - u^{k+1}\|_G^2, \tag{1.8}$$

where $u^*$ is any solution point of VI$(\Omega, F)$. Since to get the solution of (1.7) is not in general an easy matter, using the simple projection form, the prediction-correction methods [4, 6] take

$$\tilde{u}^k = P_\Omega\{u^k - \beta_k G^{-1} F(u^k)\} \tag{1.9}$$

as the predictor, and then

$$u^{k+1} = P_\Omega\{u^k - \alpha_k \beta_k G^{-1} F(\tilde{u}^k)\} \tag{1.10}$$

as the new iterate (corrector). Note that, besides the step-length $\beta_k$ and $\alpha_k$, (1.9) and (1.10) are variant forms of the projection equation (1.7) by setting the unknown vector $u$ in its right hand side by $u^k$ and $\tilde{u}^k$, respectively. The main advantage of the method (1.9)-(1.10) is that it only uses the function value of $f$ for given variable, and thus it is more practical for the problems in the real-life. However, if we use the recursion (1.9)-(1.10) to solve problem (1.4)-(1.5), sometimes the method exhibit slow convergence in practice.

The purpose of this paper is to improve the method (1.9)-(1.10) to a more effective one for solving (1.4)-(1.5). Our main improvement is twofold. Firstly, instead of the Jacobian form (1.9), we use a Gauss-Seidel form

$$\tilde{u}^k = P_\Omega\{u^k - \beta_k G^{-1} F(x^k, \tilde{y}^k)\}, \tag{1.11}$$

namely,

$$\tilde{y}^k = P_\mathcal{Y}\{y^k + \tfrac{\beta_k}{\nu}(A^T x^k - b)\} \tag{1.12a}$$

and

$$\tilde{x}^k = P_\mathcal{X}\{x^k - \beta_k[f(x^k) + A\tilde{y}^k]\}. \tag{1.12b}$$

By exploiting the structure aspect of (1.5), we decompose $u$ into two lower dimension parts $x$ and $y$ which can be obtained separately. Moreover, we use the obtained $\tilde{y}^k$ from (1.12a) rather $y^k$ in the right hand side of (1.12b) to get $\tilde{x}^k$, so we can use the new information as soon as possible. Secondly, since the solution times are significantly dependent on the



parameter $\nu$ for individual problems, how to find a proper $\nu$ to accelerate convergence is an important issue. We impose a self-adaptive rule which determine scalar $\nu$ dynamically in the iteration process.

An important and useful property of the projection mapping is

$$(v - P_\Omega(v))^T(u - P_\Omega(v)) \le 0, \quad \forall v \in \Re^l, \forall u \in \Omega. \tag{1.13}$$

**Lemma 1.1** *The VI$(\Omega, F)$ problem (1.4)-(1.5) can be equivalently solved by seeking a zero point of the mapping*

$$e(u) := \begin{pmatrix} e_x(u) \\ e_y(u) \end{pmatrix} = \begin{pmatrix} x - P_\mathcal{X}\{x - [f(x) + Ay]\} \\ y - P_\mathcal{Y}\{y + [A^T x - b]\} \end{pmatrix}. \tag{1.14}$$

Throughout this paper, we make the following standard assumptions:

**Assumption A:**

**A1**. $\mathcal{X}$ and $\mathcal{Y}$ are *simple* closed convex sets. Here a set is said to be simple means that the projection on the set is simple to carry out, for example, the positive orthant, a ball or a box.

**A2**. $f(x)$ is monotone and Lipschitz continuous with respect to $\mathcal{X}$. $L$ is the Lipschitz constant of mapping $f$.

**A3**. The solution set of VI$(\Omega, F)$, denoted by $\Omega^*$, is nonempty.

Note that $F(u)$ in (1.5) is a monotone operator with respect to $\Omega$ whenever $f(x)$ is monotone with respect to $\mathcal{X}$. The solution set $\Omega^*$ of VI$(\Omega, F)$ is closed and convex. For any $u \in \Omega$, we denote the Euclidean distance from $u$ to $\Omega^*$ by

$$\text{dist}(u, \Omega^*) := \min\{\|u - u^*\| \mid u^* \in \Omega^*\}\ .$$

It is clear that

$$\text{dist}(u, \Omega^*) = 0 \quad \Longleftrightarrow \quad e(u) = 0.$$

Historically, the term $\|e(u)\|$ is referred to as the error bound of VI$(\Omega, F)$, since it measures the magnitude of $u$ being away from $\Omega^*$.



# 2 Algorithm and remarks

In this section we review some present the new method and give some remarks. To see the procedure clearly, we present the new method in detail.

**The general version of the proposed self-adaptive prediction-correction method**

Let $\nu_0 > 0$, $\tau \in (0,1)$ and $\eta \in (0,1)$ be given constants.

**Prediction step**: For a given $u^k = (x^k, y^k) \in \mathcal{X} \times \mathcal{Y}$ and $\nu_k > 0$, set

$$\tilde{y}^k = P_{\mathcal{Y}}[y^k + \tfrac{\beta_k}{\nu_k}(A^T x^k - b)] \tag{2.1a}$$

and

$$\tilde{x}^k = P_{\mathcal{X}}\{x^k - \beta_k[f(x^k) + A\tilde{y}^k]\}. \tag{2.1b}$$

where $\beta_k$ is a proper chosen parameter which satisfies

$$\|G_k^{-1}\xi^k\|_{G_k}^2 \le \eta^2 \|u^k - \tilde{u}^k\|_{G_k}^2, \tag{2.2}$$

where

$$\xi^k = \begin{pmatrix} \xi_x^k \\ \xi_y^k \end{pmatrix} = \begin{pmatrix} \beta_k(f(\tilde{x}^k) - f(x^k)) \\ \beta_k A^T(x^k - \tilde{x}^k) \end{pmatrix} \quad \text{and} \quad G_k = \begin{pmatrix} I_n & 0 \\ 0 & \nu_k I_m \end{pmatrix}. \tag{2.3}$$

**Correction step**: Take the new iterate $u^{k+1}$, called corrector, by setting

$$x^{k+1} = P_{\mathcal{X}}\{x^k - \alpha_k \beta_k [f(\tilde{x}^k) + A\tilde{y}^k]\} \tag{2.4a}$$

and

$$y^{k+1} = P_{\mathcal{Y}}\{y^k + \alpha_k \tfrac{\beta_k}{\nu_k}(A^T\tilde{x}^k - b)\}, \tag{2.4b}$$

where

$$\alpha_k = \gamma\alpha_k^*, \qquad \alpha_k^* = \frac{(u^k - \tilde{u}^k)^T G_k d(u^k, \tilde{u}^k, \xi^k)}{\|d(u^k, \tilde{u}^k, \xi^k)\|_{G_k}^2} \tag{2.5}$$

and

$$d(u^k, \tilde{u}^k, \xi^k) := (u^k - \tilde{u}^k) + G_k^{-1}\xi^k. \tag{2.6}$$



**Adjusting $\nu$ to balance $\|\xi_x\|$ and $\|\xi_y\|/\sqrt{\nu}$ via**

$$\nu_{k+1} = \begin{cases} \nu_k * \tau, & \text{if } \ \|\xi_x^k\| \ > \ 4(\|\xi_y^k\|/\sqrt{\nu_k}), \\ \nu_k/\tau, & \text{if } \ \|\xi_y^k\|/\sqrt{\nu_k} \ > \ 4\|\xi_x^k\|, \\ \nu_k, & \text{otherwise}. \end{cases}$$

From the above description, since both prediction and correction steps make use of projections, we call the algorithm the *projection-based prediction-correction method* and abbreviate it as *P-PC* method.

**Remark**. In general, Criterion (2.2) can be satisfied via choosing a suitable $\beta_k$. For given $\nu_k > 0$, since

$$\begin{aligned} \|G_k^{-1}\xi^k\|_{G_k}^2 \quad &= \quad \beta_k^2\big(\|f(x^k) - f(\tilde{x}^k)\|^2 + \tfrac{1}{\nu_k}\|A^T(x^k - \tilde{x}^k)\|^2\big) \\ &\le \quad \beta_k^2(L^2 + \tfrac{1}{\nu_k}\|AA^T\|)\|x^k - \tilde{x}^k\|^2 \\ &\le \quad \beta_k^2(L^2 + \tfrac{1}{\nu_k}\|AA^T\|)\|u^k - \tilde{u}^k\|_{G_k}^2. \end{aligned}$$

Condition (2.2) is satisfied when

$$\beta_k \le \eta/(L^2 + \tfrac{1}{\nu_k}\|AA^T\|)^{1/2}. \tag{2.7}$$

In fact, it is enough that $f$ is locally Lipschitz continuous on a neighborhood of $x^k$.

In the correction step, $x_\alpha^{k+1}$ and $y_\alpha^{k+1}$, can be independently obtained by (2.4a) and (2.4b), the computational load of the correction step is quite tiny.

Since the error term $\|G_k^{-1}\xi^k\|_{G_k}^2 = \|\xi_x^k\|^2 + \|\xi_y^k\|^2/\nu_k$, it is reasonable to balance $\|\xi_x^k\|^2$ and $\|\xi_y^k\|^2/\nu_k$. A suitable $\nu_k > 0$ should be chosen to balance $\|\xi_x^k\|$ and $\|\xi_y^k\|/\sqrt{\nu_k}$.

# 3 Main theoretical results

Let $u^* = (x^*, y^*) \in \Omega^*$ be any solution of VI$(\Omega, F)$. Throughout this section, let $u^k = (x^k, y^k) \in \mathcal{X} \times \mathcal{Y}$ be a given vector, $\tilde{u}^k = (\tilde{x}^k, \tilde{y}^k) \in \mathcal{X} \times \mathcal{Y}$ be the predictor generated by the prediction step, and $u_\alpha^{k+1} = (x_\alpha^{k+1}, y_\alpha^{k+1}) \in \mathcal{X} \times \mathcal{Y}$ be the corrector obtained by the correction step. Let

$$\vartheta_k(\alpha) := \|u^k - u^*\|_{G_k}^2 - \|u_\alpha^{k+1} - u^*\|_{G_k}^2, \tag{3.1}$$

which measures the progress gained at the $k$-th iteration.



Note that the progress $\vartheta_k(\alpha)$ is a function of the step length $\alpha$ in the correction step. It is natural to consider maximizing this function by choosing an optimal parameter $\alpha$. The solution $u^*$, however, is unknown, so we cannot maximize $\vartheta_k(\alpha)$ directly. The main task of this section is to offer a lower bound of $\vartheta_k(\alpha)$ that does not include the unknown solution $u^*$. The following lemmas are devoted to this purpose.

**Lemma 3.1** *For given $u^k = (x^k, y^k) \in \mathcal{X} \times \mathcal{Y}$, let $\tilde{u}^k$ be the predictor produced by (2.1). For each $u' = (x', y') \in \Omega$ we have*

$$(u' - \tilde{u}^k)^T(\beta_k F(\tilde{u}^k) - \xi^k) \geq (u' - \tilde{u}^k)^T G_k(u^k - \tilde{u}^k). \tag{3.2}$$

*Particularly, if substituting $u' = u_\alpha^{k+1}$ in (3.2), we get*

$$(u_\alpha^{k+1} - \tilde{u}^k)^T \beta_k F(\tilde{u}^k) \geq (u_\alpha^{k+1} - \tilde{u}^k)^T G_k d(u^k, \tilde{u}^k, \xi^k), \tag{3.3}$$

*where $d(u^k, \tilde{u}^k, \xi^k)$ is defined in* (2.6).

**Proof**: Since $u' \in \Omega$, it follows from (2.1) and (1.13) that

$$(x' - \tilde{x}^k)^T\{\tilde{x}^k - [x^k - \beta_k(f(x^k) + A\tilde{y}^k)]\} \geq 0 \tag{3.4a}$$

$$(y' - \tilde{y}^k)^T\{\tilde{y}^k - [y^k + \tfrac{\beta_k}{\nu_k}(A^T x^k - b)]\} \geq 0. \tag{3.4b}$$

It can be rewritten as

$$(x' - \tilde{x}^k)^T\{(\tilde{x}^k - x^k) + \beta_k(f(\tilde{x}^k) + A\tilde{y}^k) - \beta_k(f(\tilde{x}^k) - f(x^k))\} \geq 0 \quad \text{(3.5a)}$$

$$(y' - \tilde{y}^k)^T\{\nu_k(\tilde{y}^k - y^k) - \beta_k(A^T\tilde{x}^k - b) - \beta_k A^T(x^k - \tilde{x}^k)\} \geq 0. \quad \text{(3.5b)}$$

Using the notation of $\xi^k$, (3.5) can be written as

$$(u' - \tilde{u}^k)^T\{G_k(\tilde{u}^k - u^k) + \beta_k F(\tilde{u}^k) - \xi^k\} \geq 0 \quad \text{(3.6)}$$

and thus the first assertion is proved. Since $u_\alpha^{k+1} \in \Omega$, by substituting $u' = u_\alpha^{k+1}$ in (3.2) we get

$$(u_\alpha^{k+1} - \tilde{u}^k)^T\beta_k F(\tilde{u}^k) \geq (u_\alpha^{k+1} - \tilde{u}^k)^T\{G_k(u^k - \tilde{u}^k) + \xi^k\}. \quad \text{(3.7)}$$

The assertion follows from $G_k(u^k - \tilde{u}^k) + \xi^k = G_k d(u^k, \tilde{u}^k, \xi^k)$ (see (2.6)) directly and the lemma is proved. $\square$

**Lemma 3.2** *For given $u^k = (x^k, y^k) \in \mathcal{X} \times \mathcal{Y}$, let $\tilde{u}^k$ be the predictor produced by (2.1) and $u_\alpha^{k+1}$ be the corrector (dependent on $\alpha$) produced by (2.4), then we have*

$$\vartheta_k(\alpha) \geq \|u^k - u_\alpha^{k+1}\|_{G_k}^2 + 2\alpha\beta_k(u_\alpha^{k+1} - \tilde{u}^k)^T F(\tilde{u}^k). \quad \text{(3.8)}$$



**Proof**: Since $u^* \in \Omega$, it follows from (2.4) and (1.13) that

$$\big(x^* - x_\alpha^{k+1}\big)^T\{x_\alpha^{k+1} - [x^k - \alpha\beta_k(f(\tilde{x}^k) + A\tilde{y}^k)]\} \geq 0 \quad \text{(3.9a)}$$

$$\big(y^* - y_\alpha^{k+1}\big)^T\{y_\alpha^{k+1} - [y^k + \alpha\tfrac{\beta_k}{\nu_k}(A^T\tilde{x}^k - b)]\} \geq 0. \quad \text{(3.9b)}$$

It follows that

$$\big(u^* - u_\alpha^{k+1}\big)^T\{G_k(u_\alpha^{k+1} - u^k) + \alpha\beta_k F(\tilde{u}^k)\} \geq 0$$

and thus

$$(u_\alpha^{k+1} - u^*)^T G_k(u^k - u_\alpha^{k+1}) \geq (u_\alpha^{k+1} - u^*)^T(\alpha\beta_k F(\tilde{u}^k)). \quad \text{(3.10)}$$

Notice that the following is an identity

$$\begin{aligned}&(u_\alpha^{k+1} - u^*)^T G_k(u^k - u_\alpha^{k+1}) \\ &= \tfrac{1}{2}\big(\|u^k - u^*\|_{G_k}^2 - \|u_\alpha^{k+1} - u^*\|_{G_k}^2\big) - \tfrac{1}{2}\|u^k - u_\alpha^{k+1}\|_{G_k}^2. \quad \text{(3.11)}\end{aligned}$$

Substituting (3.11) in (3.10) and using the definition of $\vartheta_k(\alpha)$ (see (3.1)), we get

$$\begin{aligned}\vartheta_k(\alpha) &\geq \|u^k - u_\alpha^{k+1}\|_{G_k}^2 + 2\alpha\beta_k(u_\alpha^{k+1} - u^*)^T F(\tilde{u}^k) \\ &= \|u^k - u_\alpha^{k+1}\|_{G_k}^2 + 2\alpha\beta_k\{(u_\alpha^{k+1} - \tilde{u}^k) + (\tilde{u}^k - u^*)\}^T F(\tilde{u}^k) \\ &\geq \|u^k - u_\alpha^{k+1}\|_{G_k}^2 + 2\alpha\beta_k(u_\alpha^{k+1} - \tilde{u}^k)^T F(\tilde{u}^k).\end{aligned}$$

The last inequality follows from the monotonicity of $F$ and $(\tilde{u}^k - u^*)^T F(u^*) \geq 0$. The assertion of this lemma is derived from the above inequality immediately. $\square$

Consequently, we have the following theorem.

**Theorem 3.1** *For given $u^k = (x^k, y^k) \in \mathcal{X} \times \mathcal{Y}$, let $\tilde{u}^k$ be the predictor produced by (2.1) and $u_\alpha^{k+1}$ be the corrector (dependent on $\alpha$) produced by (2.4). Then we have*

$$\vartheta_k(\alpha) \geq \|u^k - u_\alpha^{k+1}\|_{G_k}^2 + 2\alpha\big(u_\alpha^{k+1} - \tilde{u}^k\big)^T G_k d(u^k, \tilde{u}^k, \xi^k). \tag{3.12}$$

**Proof**: The assertion follows from Lemma 3.1 and Lemma 3.2 directly. $\square$

The following theorem provides a lower bound function of $\vartheta_k(\alpha)$ which is a concave quadratic function of $\alpha$.



**Theorem 3.2** *Let $\vartheta(\alpha)$ be defined by (3.1) and $d(u^k, \tilde{u}^k, \xi^k)$ be defined by (2.6). Then for any $u^* \in \Omega^*$ and $\alpha > 0$, we have*

$$\vartheta_k(\alpha) \geq \Phi_k(\alpha), \tag{3.13}$$

*where*

$$\Phi_k(\alpha) := 2\alpha(u^k - \tilde{u}^k)^T G_k d(u^k, \tilde{u}^k, \xi^k) - \alpha^2\|d(u^k, \tilde{u}^k, \xi^k)\|_{G_k}^2. \tag{3.14}$$

**Proof:** Recall that (see (3.1))

$$\vartheta_k(\alpha) = \|u^k - u^*\|_{G_k}^2 - \|u_\alpha^{k+1} - u^*\|_{G_k}^2. \tag{3.15}$$

It follows from (3.12) that

$$\begin{aligned}\vartheta_k(\alpha) &\geq 2\alpha\{(u_\alpha^{k+1} - u^k) + (u^k - \tilde{u}^k)\}^T G_k d(u^k, \tilde{u}^k, \xi^k) + \|u^k - u_\alpha^{k+1}\|_{G_k}^2 \\ &= 2\alpha(u^k - \tilde{u}^k)^T G_k d(u^k, \tilde{u}^k, \xi^k) - \alpha^2\|d(u^k, \tilde{u}^k, \xi^k)\|_{G_k}^2 \\ &\quad + \|(u^k - u_\alpha^{k+1}) - \alpha d(u^k, \tilde{u}^k, \xi^k)\|_{G_k}^2 \\ &\geq 2\alpha(u^k - \tilde{u}^k)^T G_k d(u^k, \tilde{u}^k, \xi^k) - \alpha^2\|d(u^k, \tilde{u}^k, \xi^k)\|_{G_k}^2.\end{aligned} \tag{3.16}$$

The assertion follows from the definition of $\Phi_k(\alpha)$ directly. $\square$

Note that $\vartheta_k(\alpha)$ can be regarded as the progress made by $u_\alpha^{k+1}$, and $\Phi_k(\alpha)$ is a lower bound of $\vartheta_k(\alpha)$. Therefore, it motivates us to choose such an $\alpha$ that we can reach the maximum of $\Phi_k(\alpha)$. Since it is a concave quadratic function of $\alpha$, $\Phi_k(\alpha)$ reaches its maximum at

$$\alpha_k^* = \frac{(u^k - \tilde{u}^k)^T G_k d(u^k, \tilde{u}^k, \xi^k)}{\|d(u^k, \tilde{u}^k, \xi^k)\|_{G_k}^2}. \tag{3.17}$$

This is just the same $\alpha_k^*$ in (2.5). Note that

$$\Phi_k(\alpha_k^*) = \alpha_k^*(u^k - \tilde{u}^k)^T G_k d(u^k, \tilde{u}^k, \xi^k). \tag{3.18}$$

Under Condition (2.2) we have

$$\begin{aligned} & 2(u^k - \tilde{u}^k)^T G_k d(u^k, \tilde{u}^k, \xi^k) - \|d(u^k, \tilde{u}^k, \xi^k)\|_{G_k}^2 \\ \overset{(2.6)}{=} \quad & 2\{\|u^k - \tilde{u}^k\|_{G_k}^2 + (u^k - \tilde{u}^k)^T \xi^k\} - \|(u^k - \tilde{u}^k) + G^{-1}\xi^k\|_{G_k}^2 \\ = \quad & \|u^k - \tilde{u}^k\|_{G_k}^2 - \|G^{-1}\xi^k\|_{G_k}^2 \\ \overset{(2.2)}{\geq} \quad & (1-\eta^2)\|u^k - \tilde{u}^k\|_{G_k}^2. \end{aligned} \tag{3.19}$$



Therefore, whenever $u^k \neq \tilde{u}^k$, it follows from (3.17) and (3.19) that

$$\alpha_k^* > \frac{1}{2}. \tag{3.20}$$

Consequently, from (3.18), (3.19) and (3.20) we obtain

$$\Phi_k(\alpha_k^*) \geq \tfrac{(1-\eta^2)}{4}\|u^k - \tilde{u}^k\|_{G_k}^2. \tag{3.21}$$

Considering numerical experiments, we prefer to multiply the 'optimal' value $\alpha_k^*$ by a relaxation factor $\gamma \in [1, 2)$ (better when close to 2). Thus we suggest using the correction formula (2.4) with step-size (2.5).

# 4 Convergence for the proposed method

In this section we mainly focus on investigating the convergence of the proposed method. The following theorem concerns the contractive property of the sequence generated by the proposed method.

**Theorem 4.1** *Let* $u^{k+1} = u^{k+1}(\gamma\alpha_k^*)$ *be the new iterate. Then for any* $u^* \in \Omega^*$ *and*

$\gamma \in [1, 2)$, *we have*

$$\|u^{k+1} - u^*\|_{G_k}^2 \le \|u^k - u^*\|_{G_k}^2 - \frac{\gamma(2-\gamma)(1-\eta^2)}{4}\|u^k - \tilde{u}^k\|_{G_k}^2. \tag{4.1}$$

**Proof:** We denote $\varphi_k = (u^k - \tilde{u}^k)^T G_k d(u^k, \tilde{u}^k, \xi^k)$ for convenience. Note that for $\gamma \in [1, 2)$, by a simple manipulation we obtain

$$\begin{aligned} \Phi_k(\gamma\alpha_k^*) &\overset{(3.14)}{=} 2\gamma\alpha_k^*\varphi_k - (\gamma^2\alpha_k^*)(\alpha_k^*\|d(u^k, \tilde{u}^k, \xi^k)\|_{G_k}^2) \\ &\overset{(3.17)}{=} (2\gamma\alpha_k^* - \gamma^2\alpha_k^*)\varphi_k \\ &\overset{(3.18)}{=} \gamma(2-\gamma)\Phi_k(\alpha_k^*). \end{aligned} \tag{4.2}$$

It follows from Theorem 3.2 and (4.2) that

$$\begin{aligned} \vartheta_k(\gamma\alpha^*) &= \|u^k - u^*\|^2 - \|u^{k+1}(\gamma\alpha^*) - u^*\|^2 \\ &\ge \Phi_k(\gamma\alpha^*) \\ &= \gamma(2-\gamma)\Phi_k(\alpha_k^*). \end{aligned} \tag{4.3}$$

Then the assertion follows from (3.21) immediately. $\square$



It follows from Theorem 4.1 that there is a constant $c > 0$ such that

$$\|u^{k+1} - u^*\|_{G_k}^2 \le \|u^k - u^*\|_{G_k}^2 - c\|u^k - \tilde{u}^k\|_{G_k}^2, \quad \forall u^* \in \Omega^*. \tag{4.4}$$

Therefore, the sequence $\{u^k\}$ is bounded and the proposed method belongs to contractive methods because its new iterate is closer to the solution set $\Omega^*$. Now, we are at the stage to prove the convergence of the proposed method.

**Theorem 4.2** *If* $\inf_{k=0}^{\infty} \beta_k := \beta > 0$*, then the sequence* $\{u^k\}$ *generated by the proposed method converges to some* $u^\infty$ *which is a solution of VI*$(\Omega, F)$.

**Proof:** First, using Lemma 3.1, for each $u \in \Omega$ we have

$$(u - \tilde{u}^k)^T(\beta_k F(\tilde{u}^k) - \xi^k) \ge (u - \tilde{u}^k)^T G(u^k - \tilde{u}^k). \tag{4.5}$$

It follows from (4.4) that $\{u^k\}$ is a bounded sequence and

$$\lim_{k\to\infty} \|u^k - \tilde{u}^k\|_G = 0. \tag{4.6}$$

Consequently, $\{\tilde{u}^k\}$ is also bounded. Since $\lim_{k\to\infty} \|u^k - \tilde{u}^k\|_G = 0$, $\|G^{-1}\xi^k\|_G < c_0\|u^k - \tilde{u}^k\|_G$ ($c_0$ is a positive constant) and $\beta_k \ge \beta > 0$, it follows from (4.5) that

$$\lim_{k\to\infty}(u-\tilde{u}^k)^T F(\tilde{u}^k) \geq 0, \qquad \forall u \in \Omega.$$

Because $\{\tilde{u}^k\}$ is bounded, it has at least a cluster point. Let $u^\infty$ be a cluster point of $\{\tilde{u}^k\}$ and the subsequence $\{\tilde{u}^{k_j}\}$ converges to $u^\infty$. It follows that

$$\lim_{j\to\infty}(u-\tilde{u}^{k_j})^T F(\tilde{u}^{k_j}) \geq 0, \qquad \forall u \in \Omega$$

and consequently

$$(u-u^\infty)^T F(u^\infty) \geq 0, \qquad \forall u \in \Omega.$$

This means that $u^\infty$ is a solution of VI$(\Omega, F)$. Note that the inequality(4.4) is true for all solution points of VI$(\Omega, F)$, hence we have

$$\|u^{k+1}-u^\infty\|_G^2 \leq \|u^k-u^\infty\|_G^2, \quad \forall k \geq 0. \tag{4.7}$$

Since $\tilde{u}^{k_j} \to u^\infty (j\to\infty)$ and $u^k - \tilde{u}^k \to 0 (k\to\infty)$, for any given $\varepsilon > 0$, there exists an integer $l > 0$ such that

$$\|\tilde{u}^{k_l}-u^\infty\|_G < \varepsilon/2 \quad \text{and} \quad \|u^{k_l}-\tilde{u}^{k_l}\|_G < \varepsilon/2. \tag{4.8}$$



Therefore, for any $k \geq k_l$, it follows from (4.7) and (4.8) that

$$\|u^k-u^\infty\|_G \leq \|u^{k_l}-u^\infty\|_G \leq \|u^{k_l}-\tilde{u}^{k_l}\|_G + \|\tilde{u}^{k_l}-u^\infty\|_G \leq \varepsilon.$$

This implies that $\{u^k\}$ converges to $u^\infty$ which is a solution of VI$(\Omega, F)$. $\square$

## 5 A self-adaptive version

For given $x^k \in \mathcal{X}$, the main task of the proposed method is to find a suitable $\beta_k > 0$ in the prediction step, so that (2.2) is satisfied. In particular, an acceptable $\beta_k$ can be directly obtained if $\beta_k$ is suitably small, for example, see (2.7) in Section 2. However, in practice, we suggest a self-adaptive procedure to find such a suitable $\beta_k$:

Note that the notation

$$d_x^k = x^k - \tilde{x}^k \quad \text{and} \quad d_y^k = y^k - \tilde{y}^k,$$

Condition (2.2) can be written as $r_k \le \eta$ where

$$r_k := \left( \frac{\|\xi_x^k\|^2 + \|\xi_y^k\|^2/\nu_k}{\|d_x^k\|^2 + \nu_k\|d_y^k\|^2} \right)^{1/2}. \tag{5.1}$$

For given $x^k \in \mathcal{X}$, $y \in \mathcal{Y}$, $\nu_k > 0$ and $\beta_k > 0$, we get a trial predictor $\tilde{u}^k = (\tilde{x}^k, \tilde{y}^k)$ via

$$\tilde{y}^k = P_{\mathcal{Y}}[y^k + \tfrac{1}{\nu_k}\beta_k(A^T x^k - b)]$$

and

$$\tilde{x}^k = P_{\mathcal{X}}\{x^k - \beta_k[f(x^k) + A\tilde{y}^k]\}. \tag{5.2}$$

Then calculate $r_k$ (see (5.1)). If $r_k \le \eta$, the prediction $\tilde{u}^k = (\tilde{x}^k, \tilde{y}^k)$ is accepted; otherwise, reduce the value of $\beta_k$ by $\beta_k := \beta_k * 0.8/r_k$ and repeat the procedure.

However, according to our numerical experiments, too small values of $\beta_k$ usually lead to extremely slow convergence. Thus it is necessary to avoid this situation. To do so, a strategy of enlarging $\beta_k$ is proposed in the following special version (see Step 3 for the implementation details).

It is well known that solving VI$(\Omega, F)$ (1.4)-(1.5) is equivalent to finding a zero point of $e(u)$, so we choose $\|e(u^k)\|_\infty < \varepsilon$ as the stop criterion.



### Implementation details of a special version of the QP method

Step 0. Let $\beta_0 = 1$, $\nu_0(:= 1) > 0$, $\tau(:= 0.5) > 0$, $\eta(:= 0.95) < 1$, $\gamma = 1.95$, $\varepsilon(:= 10^{-6}) > 0$, $k = 0$, $x^0 > 0$ and $y^0 \in \mathcal{Y}$.

Step 1. If $\|e(u^k)\|_\infty < \varepsilon$, then stop. Otherwise, continue.

Step 2. (Prediction step) Produce the predictor $\tilde{u}^k = (\tilde{x}^k, \tilde{y}^k)$:

Calculate $f(x^k)$ and $A^T x^k - b$;

1) $\tilde{y}^k = P_{\mathcal{Y}}\{y^k + \frac{\beta_k}{\nu_k}(A^T x^k - b)\}$;

$\tilde{x}^k := P_{\mathcal{X}}\{x^k - \beta_k[f(x^k) + A\tilde{y}^k]\}$;

$d_x^k := x^k - \tilde{x}^k$; $\quad d_y^k := y^k - \tilde{y}^k$;

$\xi_x^k := \beta_k(f(\tilde{x}^k) - f(x^k))$; $\quad \xi_y^k := \beta_k A^T(x^k - \tilde{x}^k)$;

$$r_k := \sqrt{\frac{\|\xi_x^k\|^2 + \|\xi_y^k\|^2/\nu_k}{\|d_x^k\|^2 + \nu_k\|d_y^k\|^2}};$$

2) If $r_k > \eta$, then reduce $\beta_k$ by $\beta_k := \beta_k * 0.8/r_k$, and go to 1).

Step 3. Adjust $\beta$ for the next iteration if necessary:

Prepare an enlarged $\beta$ for next iteration if $r_k$ is too small,

$$\beta_{k+1} := \begin{cases} \beta_k * 0.7/r_k & \text{if } r_k \le 0.5, \\ \beta_k & \text{otherwise,} \end{cases}$$

Step 4. Calculate the step-size in the correction step (see (2.5) and (2.6)):

$$\alpha_k^* = \frac{(d_x^k + \xi_x^k)^T d_x^k + (\nu_k \cdot d_y^k + \xi_y^k)^T d_y^k}{[d_x^k + \xi_x^k]^T [d_x^k + \xi_x^k] + (\nu_k \cdot d_y^k + \xi_y^k)^T (d_y^k + \nu_k^{-1} \xi_y^k)};$$

$$\alpha_k = \gamma \alpha_k^* \beta_k.$$

Step 5. (Correction step) Calculate the new iterate $u^{k+1} = (x^{k+1}, y^{k+1})$:

$$x^{k+1} := P_{\mathcal{X}}\{x^k - \alpha_k[f(\tilde{x}^k) + A\tilde{y}^k]\};$$
$$y^{k+1} := P_{\mathcal{Y}}\{y^k + \tfrac{\alpha_k}{\nu_k}(A^T \tilde{x}^k - b)\}.$$

Step 6. (Adjusting $\nu$ )

$$\nu_{k+1} = \begin{cases} \nu_k * \tau, & \text{if } \|\xi_x^k\| > 4(\|\xi_y^k\|/\sqrt{\nu_k}), \\ \nu_k/\tau, & \text{if } \|\xi_y^k\|/\sqrt{\nu_k} > 4\|\xi_x^k\|, \\ \nu_k, & \text{otherwise.} \end{cases}$$

$k := k + 1$; go to Step 1.

From the implementation details, we see that the entire computational cost of the proposed method is very tiny, thus the new method is applicable in practice.



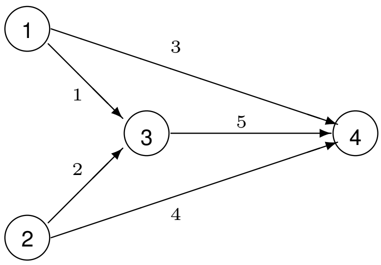
In contrast to the influence of the alteration of $\nu_k$, the adjustment of $\beta_k$ does not effect the form of contractive property (4.1). Therefore, the proof of convergence retains true when suitable $\beta_k$ is searched for in practical numerical experiments.

# 6 Application in traffic equilibrium problems

The test examples in this section are arising from the traffic equilibrium problems.

## 6.1 Traffic equilibrium problems

As an application we use the examples in the traffic equilibrium problems. The original problem is in [9] (also see [5]). Consider a network $[N, L]$ of nodes $N$ and directed links $L$, which consists of a finite sequence of connecting links with a certain orientation. Let $a, b$, etc., denote the links, and let $p, q$, etc., denote the paths. We let $\omega$ denote an origin/destination (O/D) pair of nodes of the network and $P_\omega$ denote the set of all paths connecting the O/D pair $\omega$. An illustrative example is depicted in Fig. 6.1.




| O/D Pairs | Path No. & the link on the path |
|---|---|
| $\omega_1$ : ①→ ④ | $p_1 = \{3\}$<br>$p_2 = \{1, 5\}$ |
| $\omega_2$ : ②→ ④ | $p_3 = \{4\}$<br>$p_4 = \{2, 5\}$ |

Fig. 6. 1. An example of given directed network and the O/D pairs

Let $A$ and $B$ denote the path-arc and the path-O/D pair incidence matrices, respectively.



For the given example in Fig. 6.1, $A$ and $B$ have the following forms:

$$
\begin{array}{ll}
\text{No. link} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
A = & \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}
\end{array}
\begin{array}{l} \leftarrow p_1 \\ \leftarrow p_2 \\ \leftarrow p_3 \\ \leftarrow p_4 \end{array},
\qquad
\begin{array}{ll}
\text{No. O/D pair} & \begin{array}{cc} \omega_1 & \omega_2 \end{array} \\
B = & \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}
\end{array}
\begin{array}{l} \leftarrow p_1 \\ \leftarrow p_2 \\ \leftarrow p_3 \\ \leftarrow p_4 \end{array}.
$$

Let $x_p$ represent the traffic flow on path $p$, $f_a$ denote the link load on link $a$ and $d_\omega$ denote the traffic amount between the O/D pair $\omega$. Thus the arc-flow vector $f$ is given by

$$f = A^T x$$

and the O/D pair-traffic amount vector $d$ is given by

$$d = B^T x.$$

Let $t(f) = \{t_a, a \in L\}$ be the vector of link travel costs, which is a function of the link flow. A user travelling on path $p$ incurs a (path) travel cost $\theta_p$. For given link travel cost vector $t$, the path travel cost vector $\theta$ is given by

$$\theta = At(f) \qquad \text{and thus} \qquad \theta(x) = At(A^T x).$$

Associated with every O/D pair $\omega$, there is a travel disutility $\lambda_\omega(d)$. Since both the path costs and the travel disutilities are functions of the flow pattern $x$, the traffic network equilibrium problem is to seek the path flow pattern $x^*$:

$$x^* \geq 0, \qquad (x - x^*)^T F(x^*) \geq 0, \qquad \forall x \geq 0 \tag{6.1}$$

where

$$F_p(x) = \theta_p(x) - \lambda_\omega(d(x)), \qquad \forall\, \omega, \;\; p \in P_\omega.$$

Using matrices $A$ and $B$, a compact form of mapping is
$F(x) = At(A^T x) - B\lambda(B^T x)$. The problem is a NCP. We take some test examples from [9] in which the disutility function $\lambda_\omega(d)$ is given by

$$\lambda_\omega(d) = -m_\omega d_\omega + q_\omega, \quad \forall \omega. \tag{6.2}$$

**Example**. The example (Example 7.5 in [9]) is consisted of 25 nodes, 37 links and 6 O/D pairs. The network is depicted in Figure 6.2. The user cost of traversing link $a$ is given in



Table 6.1. The O/D pairs, the coefficients $m_\omega$ and $q_\omega$ in the disutility function (6.2) and numbers of paths of each O/D pair for this problem are given in Table 6.2. Since there are together 55 paths for the 6 given O/D pairs, the dimension of the variable $x$ is 55, and the path-arc incidence matrix $A$ and the path-O/D pair incidence matrix $B$ is a $55 \times 37$ matrix and a $55 \times 6$ matrix, respectively.

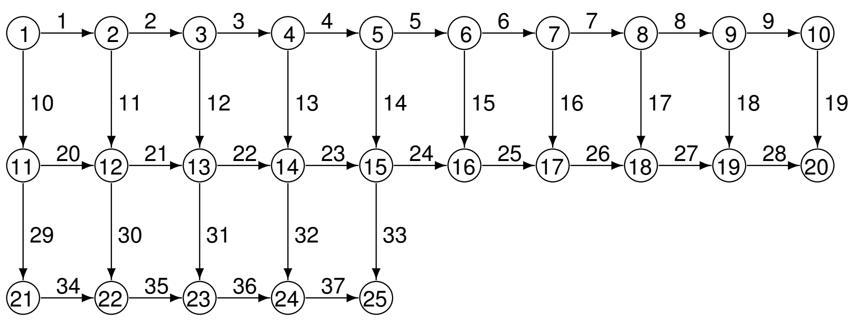


Fig. 6.2. A directed network with 25 nodes and 37 links

Table 6.1. The link traversing cost functions $t_a(f)$ in the test Example

| | |
|---|---|
| $t_1(f) = 5 \cdot 10^{-6} f_1^4 + 0.5 f_1 + 0.2 f_2 + 50$ | $t_{20}(f) = 3 \cdot 10^{-6} f_{20}^4 + 0.6 f_{20} + 0.1 f_{21} + 30$ |
| $t_2(f) = 3 \cdot 10^{-6} f_2^4 + 0.4 f_2 + 0.4 f_1 + 20$ | $t_{21}(f) = 4 \cdot 10^{-6} f_{21}^4 + 0.4 f_{21} + 0.1 f_{22} + 40$ |
| $t_3(f) = 5 \cdot 10^{-6} f_3^4 + 0.3 f_3 + 0.1 f_4 + 35$ | $t_{22}(f) = 2 \cdot 10^{-6} f_{22}^4 + 0.6 f_{22} + 0.1 f_{23} + 50$ |
| $t_4(f) = 3 \cdot 10^{-6} f_4^4 + 0.6 f_4 + 0.3 f_5 + 40$ | $t_{23}(f) = 3 \cdot 10^{-6} f_{23}^4 + 0.9 f_{23} + 0.2 f_{24} + 35$ |
| $t_5(f) = 6 \cdot 10^{-6} f_5^4 + 0.6 f_5 + 0.4 f_6 + 60$ | $t_{24}(f) = 2 \cdot 10^{-6} f_{24}^4 + 0.8 f_{24} + 0.1 f_{25} + 40$ |
| $t_6(f) = 0.7 f_6 + 0.3 f_7 + 50$ | $t_{25}(f) = 3 \cdot 10^{-6} f_{25}^4 + 0.9 f_{25} + 0.3 f_{26} + 45$ |
| $t_7(f) = 8 \cdot 10^{-6} f_7^4 + 0.8 f_7 + 0.2 f_8 + 40$ | $t_{26}(f) = 6 \cdot 10^{-6} f_{26}^4 + 0.7 f_{26} + 0.8 f_{27} + 30$ |
| $t_8(f) = 4 \cdot 10^{-6} f_8^4 + 0.5 f_8 + 0.2 f_9 + 65$ | $t_{27}(f) = 3 \cdot 10^{-6} f_{27}^4 + 0.8 f_{27} + 0.3 f_{28} + 50$ |
| $t_9(f) = 10^{-6} f_9^4 + 0.6 f_9 + 0.2 f_{10} + 70$ | $t_{28}(f) = 3 \cdot 10^{-6} f_{28}^4 + 0.7 f_{28} + 65$ |
| $t_{10}(f) = 0.4 f_{10} + 0.1 f_{12} + 80$ | $t_{29}(f) = 3 \cdot 10^{-6} f_{29}^4 + 0.3 f_{29} + 0.1 f_{30} + 45$ |
| $t_{11}(f) = 7 \cdot 10^{-6} f_{11}^4 + 0.7 f_{11} + 0.4 f_{12} + 65$ | $t_{30}(f) = 4 \cdot 10^{-6} f_{30}^4 + 0.7 f_{30} + 0.2 f_{31} + 60$ |
| $t_{12}(f) = 0.8 f_{12} + 0.2 f_{13} + 70$ | $t_{31}(f) = 3 \cdot 10^{-6} f_{31}^4 + 0.8 f_{31} + 0.1 f_{32} + 75$ |
| $t_{13}(f) = 10^{-6} f_{13}^4 + 0.7 f_{13} + 0.3 f_{18} + 60$ | $t_{32}(f) = 6 \cdot 10^{-6} f_{32}^4 + 0.8 f_{32} + 0.3 f_{33} + 65$ |
| $t_{14}(f) = 0.8 f_{14} + 0.3 f_{15} + 50$ | $t_{33}(f) = 4 \cdot 10^{-6} f_{33}^4 + 0.9 f_{33} + 0.2 f_{31} + 75$ |
| $t_{15}(f) = 3 \cdot 10^{-6} f_{15}^4 + 0.9 f_{15} + 0.2 f_{14} + 20$ | $t_{34}(f) = 6 \cdot 10^{-6} f_{34}^4 + 0.7 f_{34} + 0.3 f_{30} + 55$ |
| $t_{16}(f) = 0.8 f_{16} + 0.5 f_{12} + 30$ | $t_{35}(f) = 3 \cdot 10^{-6} f_{35}^4 + 0.8 f_{35} + 0.3 f_{32} + 60$ |
| $t_{17}(f) = 3 \cdot 10^{-6} f_{17}^4 + 0.7 f_{17} + 0.2 f_{15} + 45$ | $t_{36}(f) = 2 \cdot 10^{-6} f_{36}^4 + 0.8 f_{36} + 0.4 f_{31} + 75$ |
| $t_{18}(f) = 0.5 f_{18} + 0.1 f_{16} + 30$ | $t_{37}(f) = 6 \cdot 10^{-6} f_{37}^4 + 0.5 f_{37} + 0.1 f_{36} + 35$ |
| $t_{19}(f) = 0.8 f_{19} + 0.3 f_{17} + 60$ | |



Table 6.2. The O/D pairs and the parameters in (6.2) of the test Example

| No. of the Pair | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $(O, D)$ | $(1, 20)$ | $(1, 25)$ | $(2, 20)$ | $(3, 25)$ | $(1, 24)$ | $(11, 25)$ |
| $m_\omega$ | 0.1 | 0.6 | 1 | 0.5 | 0.7 | 0.9 |
| $q_\omega$ | 100 | 80 | 200 | 600 | 800 | 700 |
| No. of the Paths | 10 | 15 | 9 | 6 | 10 | 5 |

The problems are solved by the proposed method. We take $x^0 = (1, 1, \ldots, 1)^T$ as starting point and stop criterion is

$$\frac{\|\min\{x, F(x)\}\|_\infty}{\|\min\{x^0, F(x^0)\}\|_\infty} \le \varepsilon. \tag{6.3}$$

The number of iteration, the mapping evaluations, and the CPU time on a Notebook Computer IBM T40 for different $\varepsilon$ are reported in Table 6.3.

Table 6.3. Numerical results for different $\varepsilon$.

| No. of iterations | | | No. of $F$ evaluations | | | CPU-time |
|---|---|---|---|---|---|---|
| $10^{-6}$ | $10^{-7}$ | $10^{-8}$ | $10^{-6}$ | $10^{-7}$ | $10^{-8}$ | Average |
| 342 | 419 | 496 | 770 | 944 | 1117 | 0.14 Sec. |
| 352 | 436 | 516 | 790 | 979 | 1159 | 0.15 Sec. |

The preliminary numerical experiments tell us that solutions are obtained in a moderate number of iterations. Theoretically, the number of evaluations of the mapping $F$ per iteration is at least 2. From Table 6.3 we see that it is approximately equal to 2.2 in our test examples. This means, in order to satisfy the conditions in the prediction step, the number of trial steps is insignificant.

We modified the problem with link and demand constraints. The network is consisted of 25 nodes, 37 links and 6 O/D pairs. The proposed method is applied to solve the modified problems. We use the same notations as [5]. The traffic equilibrium problems can be described as follows:

$$(x - x^*)^T F(x^*) \geq 0, \qquad \forall x \in S. \tag{6.4}$$

For practical applications, $S$ has the following different forms:



- Traffic equilibrium problems with link capacity bound,

$$S = \{x \in R^n \mid A^T x \leq b,\ x \geq 0\}, \qquad b \ \text{ is the given link capacity vector.} \tag{6.5}$$

- Traffic equilibrium problems with link capacity bound and demand lower bound,

$$S = \{x \in R^n \mid A^T x \leq b,\ B^T x \geq d,\ x \geq 0\}. \tag{6.6}$$

It is clear that all these traffic equilibrium problems are special cases of the structured variational inequality (1.4)-(1.5).

In all test implementations we take $u^0 = (x^0, y^0)$, where $y^0 = 0$ and each element of $x^0$ equals 1. For this test problem, the stop criterion

$$\max\Big\{\frac{\|e_x(u^k)\|_\infty}{\|e_x(u^0)\|_\infty}, \|e_y(u^k)\|_\infty\Big\} \leq \varepsilon \tag{6.7}$$

for different $\varepsilon$ is reasonable.

## 6.2 Problems with link capacity bounds

The constraints set of problem with link capacity bounds is $S = \{x \in R^n \mid A^T x \le b,\ x \ge 0\}$, where $b$ is a given capacity vector. We report the numbers of iteration, the number of mapping evaluation, and the CPU time for different capacities and different $\varepsilon$ in Table 6.4.

Table 6.4. Number of It. and mapping eval. for different $\varepsilon$

| Link flow capacity | No. of iterations | | | No. of $F$ evaluations | | | CPU-time | Scaling |
|---|---|---|---|---|---|---|---|---|
| | $10^{-4}$ | $10^{-5}$ | $10^{-6}$ | $10^{-4}$ | $10^{-5}$ | $10^{-6}$ | $\varepsilon = 10^{-6}$ | factor $\nu$ |
| $b = 30$ | 116 | 145 | 172 | 242 | 300 | 354 | 0.08 Sec. | 0.0156 |
| $b = 40$ | 183 | 221 | 250 | 387 | 467 | 532 | 0.11 Sec. | 0.0156 |



Table 6.5. The optimal link flow and the toll charge on the link when $b = 40$

| Link | Flow | Charge | Link | Flow | Charge | Link | Flow | Charge | Link | Flow | Charge |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 40.00 | 0.43 | 11 | 1.85 | 0 | 21 | 40.00 | 0.11 | 31 | 11.96 | 0 |
| 2 | 38.15 | 0 | 12 | 11.96 | 0 | 22 | 40.00 | 13.66 | 32 | 40.00 | 16.42 |
| 3 | 40.00 | 16.32 | 13 | 26.19 | 0 | 23 | 26.19 | 0 | 33 | 40.00 | 13.56 |
| 4 | 13.81 | 0 | 14 | 13.81 | 0 | 24 | 0 | 0 | 34 | 26.19 | 0 |
| 5 | 0 | 0 | 15 | 0 | 0 | 25 | 0 | 0 | 35 | 28.04 | 0 |
| 6 | 0 | 0 | 16 | 0 | 0 | 26 | 0 | 0 | 36 | 40.00 | 30.13 |
| 7 | 0 | 0 | 17 | 0 | 0 | 27 | 0 | 0 | 37 | 0 | 0 |
| 8 | 0 | 0 | 18 | 0 | 0 | 28 | 0 | 0 | $--$ | $--$ | $--$ |
| 9 | 0 | 0 | 19 | 0 | 0 | 29 | 26.19 | 0 | $--$ | $--$ | $--$ |
| 10 | 40.00 | 0.11 | 20 | 40.00 | 0.18 | 30 | 1.85 | 0 | $--$ | $--$ | $--$ |

As illustrated in Section 6.1, the output vector $x$ is the path-flow, and the link flow vector is $A^T x$. In fact, $y^*$ in the output is referred to as the toll charge on the congested link. For the example with link capacity $b = 40$ we list the optimal link flow and the toll charge in Table 6.5. Indeed, the link toll charge is greater than zero if and only if the link flow reaches the capacity.

## 6.3 Problems with link capacity bounds and demand low bounds

The constraints set of problem in this subsection is
$S = \{x \in R^n \mid A^T x \le b,\ B^T x \ge d,\ x \ge 0\}$, where $b$ and $d$ are given vectors. In the test example we let each element of $b$ and $d$ equal 40 and 10, respectively. We report the numbers of iteration, the mapping evaluation, and the CPU time for different capacities and different $\varepsilon$ in Table 6.6.

The dual variable $y^*$ can be divided into two parts $y_I^*$ (to the capacity constraints $A^T x \le b$) and $y_{II}^*$ (to the demand lower bounds $B^T x \ge d$). $y_I^*$ in the output is referred to as the toll charge on the congested link, while $y_{II}^*$ represents the subsidy on the O/D pair. For the test example we list the optimal link flow and the toll charge in Table 6.7. The optimal demand and the subsidy of each O/D pair are given in Table 6.8. The outputs coincide with the optimal condition.



Table 6.6. Number of It. and mapping eval. for differnt $\varepsilon$

| Link flow capacity | Demand low bound | No. of iterations | | | No. of $F$ evaluations | | | Scaling factor $\nu$ |
|---|---|---|---|---|---|---|---|---|
| | | $10^{-4}$ | $10^{-5}$ | $10^{-6}$ | $10^{-4}$ | $10^{-5}$ | $10^{-6}$ | |
| $A^T x \le 40$ | $B^T x \ge 10$ | 318 | 400 | 470 | 678 | 850 | 995 | 0.0156 |

Table 6.7. The optimal link flow and the toll charge on the link when $b = 40$

| Link | Flow | Charge | Link | Flow | Charge | Link | Flow | Charge | Link | Flow | Charge |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 40.00 | 1.74 | 11 | 10.00 | 0 | 21 | 40.00 | 0.69 | 31 | 10.00 | 0 |
| 2 | 40.00 | 1.26 | 12 | 10.00 | 0 | 22 | 40.00 | 30.65 | 32 | 34.65 | 0 |
| 3 | 40.00 | 33.39 | 13 | 14.61 | 0 | 23 | 19.96 | 0 | 33 | 25.35 | 0 |
| 4 | 25.39 | 0 | 14 | 8.69 | 0 | 24 | 3.30 | 0 | 34 | 23.14 | 0 |
| 5 | 16.70 | 0 | 15 | 9.88 | 0 | 25 | 13.18 | 0 | 35 | 30.00 | 0 |
| 6 | 6.82 | 0 | 16 | 0 | 0 | 26 | 13.18 | 0 | 36 | 40.00 | 29.40 |
| 7 | 6.82 | 0 | 17 | 0 | 0 | 27 | 13.18 | 0 | 37 | 14.65 | 0 |
| 8 | 6.82 | 0 | 18 | 6.82 | 0 | 28 | 20.00 | 0 | – | – – | – – |
| 9 | 0 | 0 | 19 | 0 | 0 | 29 | 23.14 | 0 | – | – – | – – |
| 10 | 40.00 | 2.56 | 20 | 36.86 | 0 | 30 | 6.86 | 0 | – | – – | – – |

Table 6.8. The optimal demand and the related subsidy

| (O,D) Pair | $(1,20)$ | $(1,25)$ | $(2,20)$ | $(3,25)$ | $(1,24)$ | $(11,25)$ |
|---|---|---|---|---|---|---|
| Optimal demand | 10 | 10 | 10 | 10 | 60 | 20 |
| Subsidy | 90.9 | 73.1 | 71.0 | 2.9 | 0 | 0 |

For all test problems, the solutions are obtained in a moderate number of iterations. Thus the proposed method is effectively applicable. Theoretically, each iteration consists of at least two times of evaluation of $f(x)$. From the numerical experiments, the number of evaluations of $f$ per iteration is approximately equal to 2.25. Roughly speaking, the convergent speed of the proposed method is linear according to the numerical experiments.



# 7 Test problems in constrained quadratic programming

The second set of test examples are random generated convex quadratic programming. As we know, there are a lot of classic methods for solving quadratic programming. Our proposed method can solve the quadratic programming, but we don't mean that it can solve this problem better than those classic methods. We aim at showing the significance of adjusting the scalar parameter $\nu_k$.

Following form is a convex quadratic programming:

$$\min\{\frac{1}{2}x^T Hx - c^T x \mid x \in \Omega\}, \quad \Omega = \{x \in R^n \mid A^T x \leq b, \ x \geq 0\}, \qquad (7.1)$$

where $H \in R^{n\times n}$ is a symmetric positive definite matrix, $A \in R^{n\times m}$, $b \in R^m$ and $c \in R^n$. We construct matrices $A$ and $H$ in the test examples as follows: For given $m$, we set $n = 2m$. The elements of $A$, i. e., $a_{ij}$, is randomly given in $(-5, 5)$. The matrix $H$ is accomplished by setting

$$H := V\Sigma V^T, \qquad (7.2)$$

where

$$V = I_n - 2\frac{vv^T}{v^Tv}, \qquad \Sigma = \text{diag}(\sigma_k) \qquad \text{and} \qquad \sigma_k = \cos\frac{k\pi}{n+1} + 1.$$

In this way, $H$ is positive definite and has prescribed eigenvalues between $(0, 2)$. If $x^*$ is the solution of Problem (7.1), according to the KKT principle, there is a $0 \le y^* \in R^m$ such that

$$\begin{array}{lll} x^* \ge 0, & Hx^* + Ay^* - c \ge 0, & {x^*}^T(Hx^* + Ay^* - c) = 0, \\ y^* \ge 0, & A^Tx^* \le b, & {y^*}^T(A^Tx^* - b) = 0. \end{array} \tag{7.3}$$

Let $\xi \in R^n$ and $z \in R^m$ be random vectors whose elements are between $(-1, 1)$. We set

$$\begin{array}{ll} x^* = \max(\xi, 0) * \tau_1, & \xi^* = \max(-\xi, 0) * \tau_2, \\ y^* = \max(z, 0) * \tau_3, & z^* = \max(-z, 0) * \tau_4, \end{array}$$

where $\tau_i,\ i = 1, 2, 3, 4$, are positive parameters. By setting

$$c = Hx^* + A^Ty^* - \xi^* \qquad \text{and} \qquad b = Ax^* + z^*,$$



we constructed a test problem of (7.1) which has the known solution point $x^*$ and the optimal Lagrangian multipliers $y^*$. We tested such problems with $\tau_1 = 0.5,\ \tau_2 = 10,\ \tau_3 = 0.1, \tau_4 = 5$ and $n$ up to $1000$. For $\|e(u)\|_\infty < 10^{-6}$, the proposed method obtained the solution in a few hundred iterations with $\|x^k - x^*\|_\infty$ and $\|y^k - y^*\|_\infty$ less than $10^{-6}$. The numerical results for self-adaptive $\nu$ is listed in Table 7.1. We also report the results by using the proposed method with fixed $\nu$ in Table 7.2.

Table 7.1. The proposed method with variable parameter $\nu$

| $n$ | $m$ | Condition number of $H$ | No. of iteration | No. of fuction evaluation | CPU (Sec.) | Scaling factor $\nu$ |
|---|---|---|---|---|---|---|
| 200 | 100 | $1.6373 \cdot 10^4$ | 234 | 482 | 0.313 | 256 |
| 400 | 200 | $6.5170 \cdot 10^4$ | 245 | 498 | 0.688 | 256 |
| 600 | 300 | $1.4639 \cdot 10^5$ | 330 | 668 | 2.703 | 512 |
| 800 | 400 | $2.6003 \cdot 10^5$ | 330 | 674 | 5.640 | 512 |
| 1000 | 500 | $4.0610 \cdot 10^5$ | 271 | 552 | 7.203 | 1024 |

Table 7.2. Numerical results of different methods with fixed $\nu$=1

| $n$ | $m$ | The proposed method | | | The Extra-gradient method | | |
|---|---|---|---|---|---|---|---|
| | | No. of Iteration | No. of $F$ Evaluation | CPU (Sec.) | No. of Iteration | No. of $F$ Evaluation | CPU (Sec.) |
| 200 | 100 | 2085 | 4181 | 1.406 | 3229 | 6611 | 2.125 |
| 400 | 200 | 2400 | 4817 | 5.594 | 3463 | 6955 | 8.594 |
| 600 | 300 | 3449 | 6909 | 27.532 | 4917 | 9955 | 40.969 |
| 800 | 400 | 4739 | 9504 | 78.906 | 6669 | 13402 | 112.219 |
| 1000 | 500 | 4203 | 8418 | 109.469 | 6398 | 13002 | 175.797 |

Particular care need to taken in the different values of the $\nu$. We can observe that $\nu$ varies widely in different problems. The numerical results indicate, compared to the case with fixed $\nu$, adjusting the factor $\nu$ significantly improves both solution time and iteration number. The computation cost of the method with self-adaptive $\nu$ is only about 10% of that with fixed $\nu$. It is important to adapt $\nu$ dynamically according to different problems.



# References


[1] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Parallel and Distributed Computation*, Numerical Methods. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[2] X. L. Fu and B. S. He, Self adaptive projection based prediction correction method for constrained variational inequalities, Frontiers of Mathematics in China, 5, 3-21, 2010

[3] B. S. He, *Inexact implicit methods for monotone general variational inequalities*, Mathematical Programming, Vol. 86, 199-217, 1999.

[4] B. S. He and L-Z. Liao, *Improvements of some projection methods for monotone nonlinear variational inequalities*, Journal of Optimization Theory and Applications Vol. 112, pp. 111-128, 2002.

[5] B.S. He, L.Z. Liao and X.M. Yuan, A LQP based interior prediction-correction method for nonlinear complementarity problems, Journal of Computational Mathematics, 24, 33-44, 2006

[6] B.S. He, X.M. Yuan and J.J.Z. Zhang, *Comparison of two kinds of prediction-correction methods for monotone variational inequalities*, Computational Optimization and Applications, **27**, pp. 247-267, 2004

[7] E. N. Khobotov, Modification of the extragradient method for solving variational inequalities and certain optimization problems. USSR. Comput. Math. Phys. 27, 120–127 (1987).

[8] G. M. Korpelevich, The Extragradient Method for Finding Saddle Points and Other Problems. Ekonomika i Matematchskie Metody. 12, 747-756 (1976).

[9] A. Nagurney and D. Zhang, (1996): *Projected Dynamical Systems and Variational Inequalities with Applications*, Kluwer Academic Publishers, Boston, Dordrecht, London.

[10] M. Teboulle, *Convergence of proximal-like algorithms*, SIAM Journal on Optimization, Vol. 7, 1069-1083, 1997.

# 凸优化和单调变分不等式的收缩算法

## 第十一讲: 结构型优化的交替方向法

Alternating direction method for
separable convex programming

南京大学数学系　　何炳生
hebma@nju.edu.cn



## 1 Structured constrained convex optimization

We consider the following structured constrained convex optimization problem

$$\boxed{\min\{\theta_1(x)+\theta_2(y)\,|\,Ax+By=b,\ x\in\mathcal{X},\ y\in\mathcal{Y}\}} \qquad (1.1)$$

where $\theta_1(x):\Re^{n_1}\to\Re,\ \theta_2(y):\Re^{n_2}\to\Re$ are convex functions (but not necessary smooth), $A\in\Re^{m\times n_1}$, $B\in\Re^{m\times n_2}$ and $b\in\Re^m$, $\mathcal{X}\subset\Re^{n_1}$, $\mathcal{Y}\subset\Re^{n_2}$ are given closed convex sets. It is clear that the split feasibility problem:

$$\text{Find a point } \ x\in\mathcal{X} \ \text{ such that } \ Ax\in\mathcal{B},$$

can be formulated as a special problem of (1.1) with $\theta_1(x)=\theta_2(y)=0$. Find $(x,y)$ such that

$$\{Ax-y=0,\ x\in\mathcal{X},\ y\in\mathcal{B}\}. \qquad (1.2)$$

Let $\lambda$ be the Lagrangian multiplier for the linear constraints $Ax+By=b$ in (1.1), the Lagrangian function of this problem is

$$L(x,y,\lambda)=\theta_1(x)+\theta_2(y)-\lambda^T(Ax+By-b),$$

which is defined on $\mathcal{X} \times \mathcal{Y} \times \Re^m$. Let $(x^*, y^*, \lambda^*)$ be an saddle point of the Lagrangian function, then $(x^*, y^*, \lambda^*) \in \Omega$ and it satisfies

$$\left\{ \begin{array}{l} \theta_1(x) - \theta_1(x^*) + (x - x^*)^T(-A^T\lambda^*) \geq 0, \\ \theta_2(y) - \theta_2(y^*) + (y - y^*)^T(-B^T\lambda^*) \geq 0, \quad \forall\, (x, y, \lambda) \in \Omega, \\ (\lambda - \lambda^*)^T(Ax^* + By^* - b) \geq 0, \end{array} \right. \tag{1.3}$$

where

$$\Omega = \mathcal{X} \times \mathcal{Y} \times \Re^m.$$

By denoting

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

and

$$\theta(u) = \theta_1(x) + \theta_2(y),$$

the first order optimal condition (1.3) can be written in a compact form such as



$$\boxed{w^* \in \Omega, \ \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \geq 0, \ \forall\, w \in \Omega. \qquad (1.4)}$$

Note that the mapping $F$ is monotone. We use $\Omega^*$ to denote the solution set of the variational inequality (1.4). For convenience we use the notations

$$v = \begin{pmatrix} y \\ \lambda \end{pmatrix} \quad \text{and} \quad \mathcal{V}^* = \{(y^*, \lambda^*) \,|\, (x^*, y^*, \lambda^*) \in \Omega^*\}.$$

## Augmented Lagrangian Method to structured COP

Augmented Lagrangian Method is one of the attractive methods for nonlinear optimization as demonstrated in Chapter 17 of [21]. We try to use the Augmented Lagrangian Method to solve (1.1) and set

$$M = (A, B) \quad \text{and} \quad \mathcal{U} = \mathcal{X} \times \mathcal{Y}.$$

Now, the problem (1.1) is rewritten as

$$\min\{\theta(u) \mid Mu = b, \ u \in \mathcal{U}\}. \tag{1.5}$$

For given $\beta > 0$, the augmented Lagrangian function of (1.5) is

$$\mathcal{L}_A(u,\lambda) = \theta(u) - \lambda^T(Mu - b) + \frac{\beta}{2}\|Mu - b\|^2,$$

which defined on $\Omega = \mathcal{U} \times \Re^m$. Directly applied Augmented Lagrangian Method to the problem (1.5), the $k$-th iteration begins with $\lambda^k$, obtain

$$u^{k+1} = \text{Argmin}\{\mathcal{L}_A(u,\lambda^k) \,|\, u \in \mathcal{U}\}, \tag{1.6}$$

and then update the new iterate by

$$\lambda^{k+1} = \lambda^k - \beta(Mu^{k+1} - b). \tag{1.7}$$

Note that $u^{k+1} \in \mathcal{U}$ generated by (1.6) satisfies

$$\theta(u) - \theta(u^{k+1}) + (u - u^{k+1})^T\{-M^T\lambda^k + \beta M^T(Mu^{k+1} - b)\} \ge 0, \ \forall u \in \mathcal{U}.$$

By using (1.7) in the last inequality, we obtain

$$u^{k+1} \in \mathcal{U}, \quad \theta(u) - \theta(u^{k+1}) + (u - u^{k+1})^T\big(-M^T\lambda^{k+1}\big) \ge 0, \quad \forall u \in \mathcal{U}. \tag{1.8}$$



Combining (1.8) and (1.7), we get $w^{k+1} \in \Omega$ and for any $w \in \Omega$, it holds that

$$\theta(u) - \theta(u^{k+1}) + \begin{pmatrix} u - u^{k+1} \\ \lambda - \lambda^{k+1} \end{pmatrix}^T \left\{ \begin{pmatrix} -M^T\lambda^{k+1} \\ Mu^{k+1} - b \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{\beta}(\lambda^{k+1} - \lambda^k) \end{pmatrix} \right\} \ge 0.$$

Substituting $w = w^*$ in the above variational inequality and using the notation of $F(w)$, we get

$$\begin{aligned} &(\lambda^{k+1} - \lambda^*)^T(\lambda^k - \lambda^{k+1}) \\ &\quad \ge \quad \beta\{(w^{k+1} - w^*)^T F(w^{k+1}) + \theta(u^{k+1}) - \theta(u^*)\}. \end{aligned} \tag{1.9}$$

Using the monotonicity of $F$ and the fact

$$\theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^T F(w^*) \ge 0,$$

we derive that the right hand side of (1.9) is non-negative. Therefore, we have

$$(\lambda^{k+1} - \lambda^*)^T(\lambda^k - \lambda^{k+1}) \ge 0, \quad \forall\, \lambda^* \in \Lambda^*. \tag{1.10}$$

It follows from (1.10) that

$$\|\lambda^k - \lambda^*\|^2 = \|(\lambda^{k+1} - \lambda^*) + (\lambda^k - \lambda^{k+1})\|^2 \ge \|\lambda^{k+1} - \lambda^*\|^2 + \|\lambda^k - \lambda^{k+1}\|^2.$$

We get the nice convergence property:

$$\|\lambda^{k+1} - \lambda^*\|^2 \le \|\lambda^k - \lambda^*\|^2 - \|\lambda^k - \lambda^{k+1}\|^2.$$

**Summarize the Augmented Lagrangian Method to structured COP**

For given $\lambda^k$, $u^{k+1} = (x^{k+1}, y^{k+1})$ is the solution of the following problem

$$(x^{k+1}, y^{k+1}) = \text{Argmin}\left\{ \begin{array}{c} \theta_1(x) + \theta_2(y) - (\lambda^k)^T(Ax + By - b) \\ +\frac{\beta}{2}\|Ax + By - b\|^2 \end{array} \middle| \begin{array}{l} x \in \mathcal{X} \\ y \in \mathcal{Y} \end{array} \right\}$$

**The new iterate** $\quad \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b).$

**Convergence** $\quad \|\lambda^{k+1} - \lambda^*\|^2 \le \|\lambda^k - \lambda^*\|^2 - \|\lambda^k - \lambda^{k+1}\|^2.$

**Shortcoming** The structure property is not used !

By using the augmented Lagrangian method for the structured problem (1.5), the $k$-th iteration is from $\lambda^k$ to $\lambda^{k+1}$. The variable $u = (x, y)$ is only an intermediate variable.



# 2 Alternating Direction Method

To overcome the shortcoming the ALM for the problem (1.1), we use the alternating direction method. The main idea is splitting the subproblem (1.6) in two parts and only the $x$-part is the intermediate variable. Thus the iteration begins with $v^0 = (y^0, \lambda^0)$.

**Applied ADM to the structured COP:** $\quad (y^k, \lambda^k) \Rightarrow (y^{k+1}, \lambda^{k+1})$

First, for given $(y^k, \lambda^k)$, $x^{k+1}$ is the solution of the following problem

$$x^{k+1} = \text{Argmin}\left\{ \begin{array}{c} \theta_1(x) - (\lambda^k)^T(Ax + By^k - b) \\ +\frac{\beta}{2}\|Ax + By^k - b\|^2 \end{array} \middle| x \in \mathcal{X} \right\} \tag{2.1a}$$

Use $\lambda^k$ and the obtained $x^{k+1}$, $y^{k+1}$ is the solution of the following problem

$$y^{k+1} = \text{Argmin}\left\{ \begin{array}{c} \theta_2(y) - (\lambda^k)^T(Ax^{k+1} + By - b) \\ +\frac{\beta}{2}\|Ax^{k+1} + By - b\|^2 \end{array} \middle| y \in \mathcal{Y} \right\} \tag{2.1b}$$

$$\lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b). \tag{2.1c}$$

**Advantages** The sub-problems (2.1a) and (2.1b) are separately solved one by one.

**Remark 2.1** *The sub-problems* (2.1a) *and* (2.1b) *is equivalent to*

$$x^{k+1} = \textit{Argmin}\Big\{\theta_1(x) + \tfrac{\beta}{2}\|(Ax + By^k - b) - \tfrac{1}{\beta}\lambda^k\|^2 \big| x \in \mathcal{X}\Big\} \tag{2.2a}$$

*and*

$$y^{k+1} = \textit{Argmin}\Big\{\theta_2(y) + \tfrac{\beta}{2}\|(Ax^{k+1} + By - b) - \tfrac{1}{\beta}\lambda^k\|^2 \big| y \in \mathcal{Y}\Big\} \tag{2.2b}$$

*respectively. Note that the equation* (2.1c) *can be written as*

$$(\lambda - \lambda^{k+1})\{(Ax^{k+1} + By^{k+1} - b) + \frac{1}{\beta}(\lambda^{k+1} - \lambda^k)\} \geq 0, \ \forall \lambda \in \Re^m. \tag{2.2c}$$

**Analysis** Note that the solution of (2.1a) and (2.1b) satisfies

$$\begin{aligned} x^{k+1} \in \mathcal{X}, \quad & \theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^T \\ & \Big\{-A^T\lambda^k + \beta A^T\big(Ax^{k+1} + By^k - b\big)\Big\} \geq 0, \ \forall\, x \in \mathcal{X} \end{aligned} \tag{2.3a}$$

and

$$\begin{aligned} y^{k+1} \in \mathcal{Y}, \quad & \theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^T \\ & \Big\{-B^T\lambda^k + \beta B^T\big(Ax^{k+1} + By^{k+1} - b\big)\Big\} \geq 0, \ \forall\, y \in \mathcal{Y}, \end{aligned} \tag{2.3b}$$



respectively. Substituting $\lambda^{k+1}$ (see (2.1c)) in (2.3) (eliminating $\lambda^k$ in (2.3)), we get

$$\begin{aligned} x^{k+1} \in \mathcal{X}, \quad & \theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^T \\ & \Big\{-A^T\lambda^{k+1} + \beta A^T B(y^k - y^{k+1})\Big\} \geq 0, \ \forall\, x \in \mathcal{X}, \end{aligned} \tag{2.4a}$$

and

$$y^{k+1} \in \mathcal{Y}, \quad \theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^T\{-B^T\lambda^{k+1}\} \geq 0, \ \forall\, y \in \mathcal{Y}. \tag{2.4b}$$

For analysis convenience, we rewrite (2.4) as $u^{k+1} = (x^{k+1}, y^{k+1}) \in \mathcal{X} \times \mathcal{Y}$.

$$\begin{aligned} \theta(u) - \theta(u^{k+1}) + \begin{pmatrix} x - x^{k+1} \\ y - y^{k+1} \end{pmatrix}^T & \left\{\begin{pmatrix} -A^T\lambda^{k+1} \\ -B^T\lambda^{k+1} \end{pmatrix} + \beta \begin{pmatrix} A^T \\ B^T \end{pmatrix} B(y^k - y^{k+1}) \right. \\ & \left. + \begin{pmatrix} 0 & 0 \\ 0 & \beta B^T B \end{pmatrix} \begin{pmatrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{pmatrix} \right\} \geq 0, \ \forall\, (x, y) \in \mathcal{X} \times \mathcal{Y}. \end{aligned}$$

Combining the last inequality with (2.2c), we have $w^{k+1} \in \Omega$ and

$$\theta(u) - \theta(u^{k+1}) + \begin{pmatrix} x - x^{k+1} \\ y - y^{k+1} \\ \lambda - \lambda^{k+1} \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\lambda^{k+1} \\ -B^T\lambda^{k+1} \\ Ax^{k+1} + By^{k+1} - b \end{pmatrix} \right.$$

$$\left. +\beta \begin{pmatrix} A^T \\ B^T \\ 0 \end{pmatrix} B(y^k - y^{k+1}) + \begin{pmatrix} 0 & 0 \\ \beta B^T B & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} y^{k+1} - y^k \\ \lambda^{k+1} - \lambda^k \end{pmatrix} \right\} \geq 0, \quad (2.5)$$

for any $w \in \Omega$. The above inequality can be rewritten as $w^{k+1} \in \Omega$ and

$$\theta(u) - \theta(u^{k+1}) + (w - w^{k+1})^T F(w^{k+1}) + \beta \begin{pmatrix} x - x^{k+1} \\ y - y^{k+1} \\ \lambda - \lambda^{k+1} \end{pmatrix}^T \begin{pmatrix} A^T \\ B^T \\ 0 \end{pmatrix} B(y^k - y^{k+1})$$

$$\geq \begin{pmatrix} y - y^{k+1} \\ \lambda - \lambda^{k+1} \end{pmatrix}^T \begin{pmatrix} \beta B^T B & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} y^k - y^{k+1} \\ \lambda^k - \lambda^{k+1} \end{pmatrix}, \ \forall w \in \Omega. \quad (2.6)$$



# 3 Convergence of ADM

Based on the analysis in the last section, we have the following lemma.

**Lemma 3.1** *Let $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1}) \in \Omega$ be generated by* (2.1) *from the given $v^k = (y^k, \lambda^k)$. Then, we have*

$$(v^{k+1} - v^*)^T H(v^k - v^{k+1}) \geq (w^{k+1} - w^*)^T \eta(y^k, y^{k+1}), \quad (3.1)$$

*where*

$$\eta(y^k, y^{k+1}) = \beta \begin{pmatrix} A^T \\ B^T \\ 0 \end{pmatrix} B(y^k - y^{k+1}) \quad (3.2)$$

*and*

$$H = \begin{pmatrix} \beta B^T B & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix}. \quad (3.3)$$

**Proof**. Setting $w = w^*$ in (2.6), and using $H$ and $\eta(y^k, y^{k+1})$, we get

$$\begin{aligned}(v^{k+1} - v^*)^T H(v^k - v^{k+1})& \\ \geq \quad & (w^{k+1} - w^*)^T \eta(y^k, y^{k+1}) \\ & +\theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^T F(w^{k+1}). \end{aligned} \tag{3.4}$$

Since $F$ is monotone, it follows that

$$\begin{aligned}\theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^T F(w^{k+1})& \\ \geq \quad \theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^T F(w^*) \geq 0.&\end{aligned}$$

The last inequality is due to $w^{k+1} \in \Omega$ and $w^* \in \Omega^*$ (see (1.4)). Substituting it in (3.4), the lemma is proved. $\square$

**Lemma 3.2** *Let* $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1}) \in \Omega$ *be generated by* (2.1) *from the given* $v^k = (y^k, \lambda^k)$*. Then, we have*

$$(w^{k+1} - w^*)^T \eta(y^k, y^{k+1}) = (\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}), \tag{3.5}$$

*and*

$$(\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}) \geq 0. \tag{3.6}$$



**Proof**. By using $\eta(y^k, y^{k+1})$ (see (3.2)), $Ax^* + By^* = b$ and (2.1c), we have

$$\begin{aligned}(w^{k+1} - w^*)^T \eta(y^k, y^{k+1})& \\ = \quad & (B(y^k - y^{k+1}))^T \beta\{(Ax^{k+1} + By^{k+1}) - (Ax^* + By^*)\} \\ = \quad & (B(y^k - y^{k+1}))^T \beta(Ax^{k+1} + By^{k+1} - b) \\ = \quad & (\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}).\end{aligned}$$

Because (2.4b) is true for the $k$-th iteration and the previous iteration, we have

$$\theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^T \{-B^T \lambda^{k+1}\} \geq 0, \ \forall\, y \in \mathcal{Y}, \tag{3.7}$$

and

$$\theta_2(y) - \theta_2(y^k) + (y - y^k)^T \{-B^T \lambda^k\} \geq 0, \ \forall\, y \in \mathcal{Y}, \tag{3.8}$$

Setting $y = y^k$ in (3.7) and $y = y^{k+1}$ in (3.8), respectively, and then adding the two resulting inequalities, we get

$$(\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}) \geq 0.$$

The assertion of this lemma is proved. $\square$

Even though $H$ is positive semi-definite (see (3.3) when $B$ is not full column rank), in this lecture we use $\|v-\tilde{v}\|_H$ to denote that

$$\|v-\tilde{v}\|_H^2 = (v-\tilde{v})^T H(v-\tilde{v}) = \beta\|B(y-\tilde{y})\|^2 + \frac{1}{\beta}\|\lambda-\tilde{\lambda}\|^2.$$

**Lemma 3.3** *Let* $w^{k+1}=(x^{k+1},y^{k+1},\lambda^{k+1})\in\Omega$ *be generated by* (2.1) *from the given* $v^k=(y^k,\lambda^k)$*. Then, we have*

$$(v^{k+1}-v^*)^T H(v^k-v^{k+1}) \ge 0, \quad \forall\, v^*\in\mathcal{V}^*. \tag{3.9}$$

**Proof**. The assertion follows (3.1), (3.5) and (3.6) directly. $\Box$

**Theorem 3.1** *Let* $w^{k+1}=(x^{k+1},y^{k+1},\lambda^{k+1})\in\Omega$ *be generated by* (2.1) *from the given* $v^k=(y^k,\lambda^k)$*. Then, we have*

$$\|v^{k+1}-v^*\|_H^2 \le \|v^k-v^*\|_H^2 - \|v^k-v^{k+1}\|_H^2, \ \forall\, v^*\in\mathcal{V}^*. \tag{3.10}$$



**Proof**. By using (3.9), we have

$$\begin{aligned}
\|v^k-v^*\|_H^2 &= \|(v^{k+1}-v^*)+(v^k-v^{k+1})\|_H^2 \\
&= \|v^{k+1}-v^*\|_H^2 + 2(v^{k+1}-v^*)^T H(v^k-v^{k+1}) \\
&\quad +\|v^k-v^{k+1}\|_H^2 \\
&\ge \|v^{k+1}-v^*\|_H^2 + \|v^k-v^{k+1}\|_H^2,
\end{aligned}$$

and thus (3.10) is proved. $\Box$

The inequality (3.10) is essential for the convergence of the alternating direction method. It tells us that the alternating direction method is a contraction method. Multiplying a factor $1/\beta$, it can be written as

$$\left\|\begin{array}{c} B(y^{k+1}-y^*) \\ \frac{1}{\beta}(\lambda^{k+1}-\lambda^*)\end{array}\right\|^2 \le \left\|\begin{array}{c} B(y^{k}-y^*) \\ \frac{1}{\beta}(\lambda^{k}-\lambda^*)\end{array}\right\|^2 - \left\|\begin{array}{c} B(y^{k}-y^{k+1}) \\ \frac{1}{\beta}(\lambda^{k}-\lambda^{k+1})\end{array}\right\|^2, \ \forall\, v^*\in\mathcal{V}^*.$$

This result is included in Theorem 1 of [14] as a special case for fixed $\beta$ and $\gamma\equiv 1$.

# 4 The extended Alternating Direction Method

In the extended ADM, the $k$-th iteration begins with $(y^k, \lambda^k)$. However, we take the solution of the classical ADM as a predictor, and denote it by $(\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)$.

1. First, for given $(y^k, \lambda^k)$, $\tilde{x}^k$ is the solution of the following problem

$$\tilde{x}^k = \text{Argmin}\left\{ \begin{array}{c} \theta_1(x) - (\lambda^k)^T(Ax + By^k - b) \\ +\frac{\beta}{2}\|Ax + By^k - b\|^2 \end{array} \middle| x \in \mathcal{X} \right\} \tag{4.1a}$$

2. Then, use $\lambda^k$ and the obtained $\tilde{x}^k$, $\tilde{y}^k$ is the solution of the following problem

$$\tilde{y}^k = \text{Argmin}\left\{ \begin{array}{c} \theta_2(y) - (\lambda^k)^T(A\tilde{x}^k + By - b) \\ +\frac{\beta}{2}\|A\tilde{x}^k + By - b\|^2 \end{array} \middle| y \in \mathcal{Y} \right\} \tag{4.1b}$$

3. Finally,

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + B\tilde{y}^k - b). \tag{4.1c}$$

Based on the predictor $(\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)$, we consider how to produce the new iterate $v^{k+1} = (y^{k+1}, \lambda^{k+1})$ and drive it more close to the set $\mathcal{V}^*$.



According to the same analysis in the last section (see (2.5)) we have $(\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ and

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ -B^T\tilde{\lambda}^k \\ A\tilde{x}^k + B\tilde{y}^k - b \end{pmatrix} + \beta \begin{pmatrix} A^T \\ B^T \\ 0 \end{pmatrix} B(y^k - \tilde{y}^k) \right.$$
$$\left. + \begin{pmatrix} 0 & 0 \\ \beta B^T B & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} \tilde{y}^k - y^k \\ \tilde{\lambda}^k - \lambda^k \end{pmatrix} \right\} \geq 0, \ \forall w \in \Omega. \tag{4.2}$$

Based on the above analysis, we have the following lemma.

**Lemma 4.1** *Let $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ be generated by (4.1) from the given $v^k = (y^k, \lambda^k)$. Then, we have*

$$(\tilde{v}^k - v^*)^T H(v^k - \tilde{v}^k) \geq (\tilde{w}^k - w^*)^T \eta(y^k, \tilde{y}^k), \tag{4.3}$$

*where*

$$\eta(y^k, \tilde{y}^k) = \beta \begin{pmatrix} A^T \\ B^T \\ 0 \end{pmatrix} B(y^k - \tilde{y}^k) \tag{4.4}$$

*and* $H$ *is the same as defined in* (3.3).

**Proof**. The prof is similar as those for Lemma 3.1 and thus omitted. $\square$

Similarly as in (3.5), by using $\eta(y^k, \tilde{y}^k)$ (see (4.4)) and $Ax^* + By^* = b$, we have

$$(\tilde{w}^k - w^*)^T \eta(y^k, \tilde{y}^k) = (\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k). \tag{4.5}$$

**Lemma 4.2** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (4.1) *from the given* $v^k = (y^k, \lambda^k)$. *Then, we have*

$$(v^k - v^*)^T H(v^k - \tilde{v}^k) \geq \varphi(v^k, \tilde{v}^k), \quad \forall\, v^* \in \mathcal{V}^*. \tag{4.6}$$

*and*

$$\varphi(v^k, \tilde{v}^k) = \|v^k - \tilde{v}^k\|_H^2 + (\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k). \tag{4.7}$$



**Proof**. It follows from (4.3) and (4.5) that

$$(\tilde{v}^k - v^*)^T H(v^k - \tilde{v}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k).$$

Assertion (4.6) follows from the last inequality and the definition of $\varphi(v^k, \tilde{v}^k)$ and the lemma is proved. $\square$

Now, we observe the right hand side of (4.6). Note that

$$\begin{aligned} \varphi(v^k, \tilde{v}^k) &= \|v^k - \tilde{v}^k\|_H^2 + (\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k) \\ &= \frac{1}{2}\|v^k - \tilde{v}^k\|_H^2 + \frac{1}{2\beta}\|\beta B(y^k - \tilde{y}^k) + (\lambda^k - \tilde{\lambda}^k)\|^2 \\ &\geq \frac{1}{2}\|v^k - \tilde{v}^k\|_H^2. \end{aligned} \tag{4.8}$$

## Ye-Yuan's Alternating Direction Method

Ye-Yuan's alternating direction method is a prediction-correction method. The predictor is generated by (4.1). The correction step is to update the new iterate.

**Correction** Using the $\tilde{v}^k$ produced by (4.1), update the new iterate $v^{k+1}$ by

$$v^{k+1} = v^k - \alpha_k(v^k - \tilde{v}^k), \qquad \alpha_k = \gamma\alpha_k^*, \qquad \gamma \in (0,2) \tag{4.9a}$$

where

$$\alpha_k^* = \frac{\varphi(v^k, \tilde{v}^k)}{\|v^k - \tilde{v}^k\|_H^2} \tag{4.9b}$$

**Usually, in comparison with the computational load for obtaining $(\tilde{x}^k, \tilde{y}^k)$ in (4.1), the calculation cost for step-size $\alpha_k^*$ is slight.**

We obtain an essential inequality for convergence in the following theorem which was proved by Ye and Yuan in [29].

**Theorem 4.1** *Let $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ be generated by (4.1) from the given $v^k = (y^k, \lambda^k)$ and the new iterate $v^{k+1}$ be given by (4.9). Then we have*

$$\|v^{k+1} - v^*\|_H^2 \le \|v^k - v^*\|_H^2 - \frac{\gamma(2-\gamma)}{4}\|v^k - \tilde{v}^k\|_H^2, \; \forall v^* \in \mathcal{V}^*. \tag{4.10}$$



**Proof**. By using (4.6) and (4.9), we obtain

$$\begin{aligned}
&\|v^k - v^*\|_H^2 - \|v^{k+1} - v^*\|_H^2 \\
&\quad = \|v^k - v^*\|_H^2 - \|(v^k - v^*) - \gamma\alpha_k^*(v^k - \tilde{v}^k)\|_H^2 \\
&\quad \ge 2\gamma\alpha_k^*\varphi(v^k, \tilde{v}^k) - \gamma^2(\alpha_k^*)^2\|v^k - \tilde{v}^k\|_H^2 \\
&\quad = \gamma(2-\gamma)(\alpha_k^*)^2\|v^k - \tilde{v}^k\|_H^2.
\end{aligned} \tag{4.11}$$

In addition, it follows from (4.8) and (4.9b) that $\alpha_k^* \ge \frac{1}{2}$. Substituting this fact in (4.11), the theorem is proved. $\square$

**Convergence** Both (3.10) and (4.10) can be written as

$$\left\|\begin{matrix} B(y^{k+1} - y^*) \\ \frac{1}{\beta}(\lambda^{k+1} - \lambda^*) \end{matrix}\right\|^2 \le \left\|\begin{matrix} B(y^k - y^*) \\ \frac{1}{\beta}(\lambda^k - \lambda^*) \end{matrix}\right\|^2 - c_0\left\|\begin{matrix} B(y^k - \tilde{y}^k) \\ \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k) \end{matrix}\right\|^2, \; \forall v^* \in \mathcal{V}^*.$$

It leads to that

$$\lim_{k\to\infty} By^k = By^* \qquad \text{and} \qquad \lim_{k\to\infty} \lambda^k = \lambda^*.$$

# 5 Application and Numerical Experiments

## 5.1 Calibrating the correlation matrices

We consider to solve the following problem:

$$\min\{\frac{1}{2}\|X - C\|_F^2 \mid X \in S_+^n \cap S_B\}, \tag{5.1}$$

where

$$S_+^n = \{H \in R^{n\times n} \mid H^T = H, \; H \succeq 0\},$$

and

$$S_B = \{H \in R^{n\times n} \mid H^T = H, \; H_L \le H \le H_U\}.$$

$H_L$ and $H_U$ are given symmetric matrices.

**Use the following Matlab Code to produce the matrices $C$, $H_L$ and $H_U$**

```
rand('state',0);     C=rand(n,n);  C=(C'+C)-ones(n,n) + eye(n);
%%% C is symmetric and C_{ij} is in (-1,1),  C_{jj} is in (0,2)  %%
HU=ones(n,n)*0.1;  HL=-HU; for i=1:n   HU(i,i)=1;  HL(i,i)=1; end;
```

**The problem is converted to the following equivalent one:**



$$\begin{array}{rl} \min & \frac{1}{2}\|X - C\|^2 + \frac{1}{2}\|Y - C\|^2 \\ \text{s.t} & X - Y = 0, \\ & X \in S_+^n, \; Y \in S_B. \end{array} \tag{5.2}$$

**The basic sub-problems in alternating direction methods for the problem (5.2)**

- For fixed $Y^k$ and $Z^k$,

$$\tilde{X}^k = \text{Argmin}\{\frac{1}{2}\|X - C\|_F^2 - \mathbf{Tr}(Z^k X) + \frac{\beta}{2}\|X - Y^k\|_F^2 \mid X \in S_+^n\}$$

- With fixed $\tilde{X}^k$ and $Z^k$,

$$\tilde{Y}^k = \text{Argmin}\{\frac{1}{2}\|Y - C\|_F^2 + \mathbf{Tr}(Z^k Y) + \frac{\beta}{2}\|\tilde{X}^k - Y\|_F^2 \mid Y \in S_B\}$$

**$\tilde{X}^k$ can be directly obtained via**

$$\tilde{X}^k = P_{S_+^n}\{\frac{1}{1+\beta}(\beta Y^k + Z^k + C)\}. \tag{5.3}$$

Note that

$$P_{S^n_+}(A) = U\Lambda^+U^T, \text{ where } \Lambda^+ = \max(\Lambda, 0) \text{ and } [U,\Lambda] = \text{eig}(A).$$

**Similarly, $\tilde{Y}^k$ in is given by**

$$\tilde{Y}^k = P_{S_B}\{\frac{1}{1+\beta}(\beta\tilde{X}^k - Z^k + C)\}. \quad (5.4)$$

$S_B = \{H \,|\, H_L \le H \le H_U\}, \qquad P_{S_B}(A) = \min(\max(H_L, A), H_U)$

**The most time consuming calculation is $[U,\Lambda] =$ eig$(A)$, $9n^3$**

The main Matlab Code of an iteration in the Classical ADM

```
Y0=Y;                              Z0=Z;          k=k+1;
X=(Y0*beta+Z0+C)/(1+beta);
   [V,D]=eig(X);                   D=max(0,D);    X=(V*D)*V';
Y=min(max((X*beta-Z0+C)/(1+beta),HL),HU);
Z=Z0-(X-Y)*beta;
```



The main Matlab Code of an iteration in Ye-Yuan's ADM

```
Y0=Y;                              Z0=Z;          k=k+1;
X=(Y0*beta+Z0+C)/(1+beta);
[V,D]=eig(X);                      D=max(0,D);    X=(V*D)*V';
Y=min(max((X*beta-Z0+C)/(1+beta),HL),HU);
%%%%%%%%%    Calculating the step size   %%%%%%%%%%%%%%
EY=Y0-Y;        EZ=(X-Y)*beta;
T1 = EY(:)'*EY(:);  T2 = EZ(:)'*EZ(:);   TA=T1*beta + T2/beta
T2 = (EY(:)'*EZ(:));
alpha=(TA-T2)*gammaY/TA;
Y=Y0-EY*alpha;                                Z=Z0-EZ*alpha;
```

**Numerical results for problem (5.1)**

C=rand(n,n);  C=(C'+C)-ones(n,n) + eye(n)

$H_U$=ones(n,n)*0.2; $\quad H_L = -H_U; \quad H_U(jj) = H_L(jj)$=1.

Numerical Results for calibrating correlation matrix (Using Matlab EIG)

| $n \times n$ Matrix | Classical ADM | | Glowinski's ADM | | Ye-Yuan's ADM | |
|---|---|---|---|---|---|---|
| $n =$ | No. It | CPU Sec. | No. It | CPU Sec. | No. It | CPU Sec. |
| 500 | 40 | 18.03 | 39 | 17.68 | 32 | 14.99 |
| 800 | 41 | 73.28 | 39 | 70.00 | 33 | 60.80 |
| 1000 | 43 | 141.69 | 42 | 138.30 | 34 | 114.67 |
| 1500 | 47 | 471.77 | 45 | 452.22 | 41 | 419.70 |
| 2000 | 55 | 1254.01 | 53 | 1206.94 | 45 | 1035.38 |

Numerical Results for calibrating correlation matrix (Using MeXEIG)

| $n \times n$ Matrix | Classical ADM | | Glowinski's ADM | | Ye-Yuan's ADM | |
|---|---|---|---|---|---|---|
| $n =$ | No. It | CPU Sec. | No. It | CPU Sec. | No. It | CPU Sec. |
| 500 | 40 | 5.57 | 39 | 5.38 | 32 | 4.67 |
| 800 | 41 | 18.13 | 39 | 17.15 | 33 | 15.13 |
| 1000 | 43 | 34.75 | 42 | 34.00 | 34 | 28.50 |
| 1500 | 47 | 123.77 | 45 | 117.87 | 41 | 110.17 |
| 2000 | 55 | 306.32 | 53 | 294.75 | 45 | 255.72 |



$$\frac{\text{It. No. of Ye-Yuan's ADM}}{\text{It. No. of Classical ADM}} \approx \frac{5}{6}.$$

It seems that Ye-Yuan's Algorithm converges faster than primary ADM.

## 5.2 Application for Sparse Covariance Selection

For the details of applications in this subsection, please see the reference [30].

**The problem**

$$\min_X \{\mathbf{Tr}(\Sigma X) - \log(\det(X)) + \rho e^T|X|e \;\big|\; X \in S^n_+\}, \tag{5.5}$$

**The equivalent problem:**

$$\begin{array}{rl} \min & \mathbf{Tr}(\Sigma X) - \log(\det(X)) + \rho e^T|Y|e \\ \text{s.t} & X - Y = 0, \\ & X \in S^n_+. \end{array} \tag{5.6}$$

## For given $Y^k$ and $Z^k$, get $(\tilde{X}^k, \tilde{Y}^k, \tilde{Z}^k)$ in the following procedure:

1. For fixed $Y^k$ and $Z^k$, $\tilde{X}^k$ is the solution of the following problem
$$\min\{\mathbf{Tr}(\Sigma X) - \log(\det(X)) - \mathbf{Tr}(Z^k X) + \frac{\beta}{2}\|X - Y^k\|_F^2 \mid X \in S_+^n\}$$
2. Then, with fixed $(\tilde{X}^k, Z^k)$, $\tilde{Y}^k$ is a solution of
$$\min\{\rho e^T |Y| e + \mathbf{Tr}(Z^k Y) + \frac{\beta}{2}\|\tilde{X}^k - Y\|_F^2\}$$
3. Finally, update $\tilde{Z}^k$ by
$$\tilde{Z}^k = Z^k - \beta(\tilde{X}^k - \tilde{Y}^k). \tag{5.7}$$

### Solving the $X$ subproblem for getting $\tilde{X}^k$:

$$\tilde{X}^k = \text{Argmin}\Big\{\frac{1}{2}\|X - [Y^k - \frac{1}{\beta}(\Sigma - Z^k)]\|_F^2 - \frac{1}{\beta}\log(\det(X)) \;\Big|\; X \in S_+^n\Big\}. \tag{5.8}$$



It should hold that $\tilde{X} \succ 0$ and thus $\tilde{X}$ is the solution of matrix equation
$$X - \Big(Y^k - \frac{1}{\beta}(\Sigma - Z^k)\Big) - \frac{1}{\beta}X^{-1} = 0. \tag{5.9}$$

By setting
$$A = Y^k - \frac{1}{\beta}(\Sigma - Z^k), \tag{5.10}$$
and using
$$[V, \Lambda] = \text{eig}(A) \tag{5.11}$$
in Matlab, we get
$$A = V\Lambda V^T, \qquad \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n).$$

In fact, the solution of matrix equation (5.9) should have the same eigenvectors as matrix $A$.
$$\tilde{X} = V\tilde{\Lambda}V^T, \qquad \tilde{\Lambda} = \text{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n). \tag{5.12}$$

It follows from (5.9) that
$$\tilde{\Lambda} - \Lambda - \frac{1}{\beta}\tilde{\Lambda}^{-1} = 0$$

and thus

$$\tilde{\lambda}_j = \frac{\lambda_j + \sqrt{\lambda_j^2 + (4/\beta)}}{2}, \quad j = 1, \ldots, n.$$

Indeed, $\tilde{\lambda}_j > 0$ and thus $\tilde{X} \succ 0$ (see (5.12)).

The main computational load for getting $\tilde{X}^k$ is the eigenvalues-vectors decomposition in (5.11).

**Solving the $Y$ subproblem for getting $\tilde{Y}^k$:**

The first order condition for minimization problem

$$\min\{\rho e^T |Y| e + \mathbf{Tr}(Z^k Y) + \frac{\beta}{2}\|\tilde{X}^k - Y\|_F^2\}$$

is

$$0 \in \frac{\rho}{\beta}\partial(|Y|) + Y - (\tilde{X}^k - \frac{1}{\beta}Z^k).$$

In fact,

$$\tilde{Y}^k = (\tilde{X}^k - \frac{1}{\beta}Z^k) - P_{B_\infty^{\rho/\beta}}[\tilde{X}^k - \frac{1}{\beta}Z^k],$$



where $B_\infty^{\rho/\beta} = \{X \in \mathbf{R}^{n\times n} \,|\, -\frac{\rho}{\beta} \le X_{ij} \le \frac{\rho}{\beta}\}$. **The projection on a 'box' is very easy to be carried out !**

## 5.3 Split feasibility problem and Matrix completion

Applying ADMM to the reformulated split feasibility problem (1.2), the $k$-th iteration begins with $(y^k, \lambda^k) \in \mathcal{B} \times \Re^m$, and the new iterate is generated by the following procedure:

$$\begin{cases} x^{k+1} = \text{Argmin}\{\frac{1}{2}\|Ax - (y^k + \lambda^k/\beta)\|^2 \,|\, x \in \mathcal{X}\}, \\ y^{k+1} = P_{\mathcal{B}}[Ax^{k+1} - \lambda^k/\beta], \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} - y^{k+1}). \end{cases}$$

The $x$-subproblem is a problem of form $\min\{\frac{1}{2}\|Ax - a\|^2 \,|\, x \in \mathcal{X}\}$. After getting $x^{k+1}$, $(y^{k+1}, \lambda^{k+1})$ is obtained by a projection and an evaluation.

Matrix completion is to recover an unknown matrix from a sampling of its entries. For an $m \times n$ matrix $M$, $\Omega$ denotes the indices subset of the matrix

$$\Omega = \{(ij) \,|\, i \in \{1, 2, \ldots, m\}, \; j \in \{1, 2, \ldots, n\}\}.$$

The mathematical form of the considered problem is

$$\boxed{\min\{\|X\|_* \mid X_{ij} = M_{ij},\ (ij) \in \Omega\}}$$

where $\|X\|_*$ is the nuclear norm of $X$ [2]. It can convert to the following problem

$$\begin{array}{ll} \min_{X,Y} & \|X\|_* \\ \textbf{s. t} & X - Y = 0, \\ & Y_{ij} = M_{ij}, \forall\,(ij) \in \Omega. \end{array}$$

It belongs to the problem (1.1) and was successfully solved by the alternating direction methods [4].

**Simple iterative scheme $+$ nice convergence properties**
**$\Rightarrow$ wide applications of ADMM in large scale optimization**



# References


[1] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Foundations and Trends in Machine Learning, **3**, 1-122, 2010.

[2] J. F. Cai, E. J. Candès and Z. W. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. **20**, 1956-1982, 2010.

[3] R. H. Chan, J. F. Yang, and X. M. Yuan. Alternating direction method for image inpainting in wavelet domain. *SIAM J. Imag. Sci.*, 4:807–826, 2011.

[4] C. H. Chen, B.S. He and X. M. Yuan, Matrix completion via alternating direction methods, IMA Journal of Numerical Analysis, **3**, 227-245, 2012.

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

[6] D. Gabay, Applications of the Method of Multipliers to Variational Inequalities, in *Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems*, M. Fortin and R. Glowinski, eds., North Holland, Amsterdam, The Netherlands, 299-331, 1983.

[7] D. Gabay and B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximations, Computers and Mathematics with Applications, **2**, 17-40, 1976.

[8] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[9] D. Goldfarb, S.Q. Ma, and K. Scheinberg. Fast alternating linearization methods for minimizing the sum of two convex functions. *Math. Program.*, (2012). DOI 10.1007/s10107-012-0530-2

[10] B.S. He, Inexact implicit methods for monotone general variational inequalities, Math. Program. Series A, **86**, 199-217, 1999.

[11] B.S. He, L-Z Liao, D.R. Han and H. Yang, A new inexact alternating directions method for monotone variational inequalities, Math. Progr., **92**, 103-118, 2002.

[12] B. S. He, M. Tao, and X. M. Yuan. Alternating direction method with Gaussian back substitution for separable convex programming. *SIAM J. Opt.*, 22(2): 313–340, 2012.

[13] B. S. He, M. H. Xu and X. M. Yuan, Solving large-scale least squares covariance matrix problems by alternating direction methods, SIAM J. Matrix Analy. Appli., 32: 136-152, 2011.

[14] B. S. He and H. Yang, Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities, Oper. Res. Let., **23**, 151-161, 1998.

[15] B.S. He, H. Yang and S.L. Wang, Alternating directions method with self-adaptive penalty parameters for monotone variational inequalities, JOTA **106**, 349-368, 2000.

[16] B.S. He and X.M. Yuan, On the $O(1/n)$ Convergence Rate of the Douglas-Rachford Alternating Direction Method, SIAM J. Numer. Anal. 50, 700-709, 2012.

[17] Z. Lu, T. K. Pong, and Y. Zhang. An alternating direction method for finding Dantzig selectors. Preprint, 2010.

[18] M. Ng, F. Wang, and X. M. Yuan. Inexact alternating direction methods for image recovery. *SIAM J. Sci. Comput.*, 33:1643–1668, 2011.

[19] M. Ng, P. A. Weiss, and X. M. Yuan. Solving constrained total-variation problems via alternating direction methods. *SIAM J. Sci. Comput.*, 32:2710–2736, 2010.

[20] S. Setzer. Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage. *Scale space and variational methods in computer vision*, 5567:464–476, 2009.



[21] J. Nocedal and S.J. Wright, Numerical Optimization, Springer Verlag, 1999.

[22] S. Setzer. Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage. *Scale space and variational methods in computer vision*, 5567:464–476, 2009.

[23] J. Sun and S. Zhang. A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs. *European J. Oper. Res*, 207:1210–1220, 2010.

[24] M. Tao and X. M. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM J. Opt.*, 21(1):57–81, 2011.

[25] Z. W. Wen, D. Goldfarb, and W. T. Yin. Alternating direction augmented Lagrangian methods for semidefinite programming. *Math. Program. Comput.*, 2:203–230, 2010.

[26] J. F. Yang and Y. Zhang. Alternating direction algorithms for $L_1$-problems in compressive sensing, *SIAM J. Sci. Comput.,* 38:250–278, 2011

[27] J. F. Yang, Y. Zhang, and W. Yin. A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data. *IEEE J. Selected Topics in Signal Processing*, 4:288–297, 2010

[28] J. F. Yang and X. M. Yuan. Sparese and low-rank matrix decomposition via alternating direction methods. *manuscript*, 2012.

[29] C. H. Ye and X. M. Yuan, A Descent Method for Structured Monotone Variational Inequalities, Optimization Methods and Software, **22**, 329-338, 2007.

[30] X. M. Yuan. Alternating direction method of multipliers for covariance selection models. *J. Sci. Comput.*, 51:261-273, 2012.

[31] X. Q. Zhang, M. Burger, and S. Osher. A unified primal-dual algorithm framework based on Bregman iteration. *J. Sci. Comput.*, 46(1):20–46, 2011.

# 凸优化和单调变分不等式的收缩算法

## 第十二讲: 线性化的交替方向收缩算法

Linearized alternating direction method
for separable convex programming

南京大学数学系 何炳生
hebma@nju.edu.cn



## 1 Structured constrained convex optimization

We consider the following structured constrained convex optimization problem

$$\min\{\theta_1(x)+\theta_2(y)\,|\,Ax+By=b,\ x\in\mathcal{X},\ y\in\mathcal{Y}\} \quad (1.1)$$

where $\theta_1(x):\Re^{n_1}\rightarrow\Re,\ \theta_2(y):\Re^{n_2}\rightarrow\Re$ are convex functions (but not necessary smooth), $A\in\Re^{m\times n_1}$, $B\in\Re^{m\times n_2}$ and $b\in\Re^m$, $\mathcal{X}\subset\Re^{n_1}$, $\mathcal{Y}\subset\Re^{n_2}$ are given closed convex sets. We let $n=n_1+n_2$.

The task of solving the problem (1.1) is to find an $(x^*,y^*,\lambda^*)\in\Omega$, such that

$$\left\{\begin{array}{l}\theta_1(x)-\theta_1(x^*)+(x-x^*)^T(-A^T\lambda^*)\ge 0,\\ \theta_2(y)-\theta_2(y^*)+(y-y^*)^T(-B^T\lambda^*)\ge 0, \quad \forall\,(x,y,\lambda)\in\Omega,\\ (\lambda-\lambda^*)^T(Ax^*+By^*-b)\ge 0,\end{array}\right. \quad (1.2)$$

where

$$\Omega=\mathcal{X}\times\mathcal{Y}\times\Re^m.$$

By denoting

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

and

$$\theta(u) = \theta_1(x) + \theta_2(y),$$

the first order optimal condition (1.2) can be written in a compact form such as

$$\boxed{w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \geq 0, \ \forall\, w \in \Omega. \quad (1.3)}$$

Note that the mapping $F$ is monotone. We use $\Omega^*$ to denote the solution set of the variational inequality (1.3). For convenience we use the notations

$$v = \begin{pmatrix} y \\ \lambda \end{pmatrix} \qquad \text{and} \qquad \mathcal{V}^* = \{(y^*, \lambda^*) \,|\, (x^*, y^*, \lambda^*) \in \Omega^*\}.$$



Alternating Direction Method is a simple but powerful algorithm that is well suited to distributed convex optimization [1]. This approach also has the benefit that one algorithm could be flexible enough to solve many problems.

**Applied ADM to the structured COP:** $(y^k, \lambda^k) \Rightarrow (y^{k+1}, \lambda^{k+1})$

First, for given $(y^k, \lambda^k)$, $x^{k+1}$ is the solution of the following problem

$$\boxed{x^{k+1} = \text{Argmin}\Big\{\theta_1(x) + \tfrac{\beta}{2}\|Ax + By^k - b - \tfrac{1}{\beta}\lambda^k\|^2 \Big| x \in \mathcal{X}\Big\} \quad (1.4a)}$$

Use $\lambda^k$ and the obtained $x^{k+1}$, $y^{k+1}$ is the solution of the following problem

$$\boxed{y^{k+1} = \text{Argmin}\Big\{\theta_2(y) + \tfrac{\beta}{2}\|Ax^{k+1} + By - b - \tfrac{1}{\beta}\lambda^k\|^2 \Big| y \in \mathcal{Y}\Big\} \quad (1.4b)}$$

$$\lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b). \qquad (1.4c)$$

In some structured convex optimization (1.1), $B$ is a scalar matrix. However, the solution of the subproblem (1.4a) does not have the closed form solution because of the general structure of the matrix $A$. In this case, we linearize the quadratic term of (1.4a)

$$\frac{\beta}{2}\|Ax + By^k - b - \tfrac{1}{\beta}\lambda^k\|^2$$

at $x^k$ and add a proximal term $\frac{r}{2}\|x - a\|^2$ to the objective function. In other words, instead of (1.4a), we solve the following $x$ subproblem:

$$\min\{\theta_1(x) + \beta x^T A^T (Ax^k + By^k - b - \tfrac{1}{\beta}\lambda^k) + \tfrac{r}{2}\|x - x^k\|^2 \,\big|\, x \in \mathcal{X}\}.$$

Based on linearizing the quadratic term of (1.4a), in this lecture, we construct the linearized alternating direction method. We still assume that the solution of the problem

$$\min\{\theta_1(x) + \frac{r}{2}\|x - a\|^2 \,|\, x \in \mathcal{X}\} \tag{1.5}$$

has a closed form.



# 2 Linearized Alternating Direction Method

In the Linearized ADM, $x$ is not an intermediate variable. The $k$-th iteration of the Linearized ADM is from $(x^k, y^k, \lambda^k)$ to $(x^{k+1}, y^{k+1}, \lambda^{k+1})$.

## 2.1 Linearized ADM

1. First, for given $(x^k, y^k, \lambda^k)$, $x^{k+1}$ is the solution of the following problem

$$x^{k+1} = \text{Argmin}\left(\begin{array}{c}\{\theta_1(x) + \beta x^T A^T (Ax^k + By^k - b - \frac{1}{\beta}\lambda^k) \\ + \frac{r}{2}\|x - x^k\|^2 \quad \big|\, x \in \mathcal{X}\}\end{array}\right). \tag{2.1a}$$

2. Then, use $\lambda^k$ and the obtained $x^{k+1}$, $y^{k+1}$ is the solution of the following problem

$$y^{k+1} = \text{Argmin}\left\{\theta_2(y) + \frac{\beta}{2}\|Ax^{k+1} + By - b - \frac{1}{\beta}\lambda^k\|^2 \Big| y \in \mathcal{Y}\right\}. \tag{2.1b}$$

3. Finally,

$$\lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b). \tag{2.1c}$$

**Requirements on parameters $\beta$, $r$** For given $\beta > 0$, choose $r$ such that

$$rI_n - \beta A^T A \succeq 0. \tag{2.2}$$

**Analysis of the optimal conditions of subproblems in (2.1)**

Note that $x^{k+1}$, the solution of (2.1a), satisfies

$$\begin{aligned} x^{k+1} \in \mathcal{X}, \quad & \theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^T \Big\{ -A^T \lambda^k \\ & + \beta A^T \big( Ax^k + By^k - b \big) + r(x^{k+1} - x^k) \Big\} \geq 0, \ \forall\, x \in \mathcal{X}. \end{aligned} \tag{2.3a}$$

Similarly, the solution of (2.1b) $y^{k+1}$ satisfies

$$\begin{aligned} y^{k+1} \in \mathcal{Y}, \quad & \theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^T \Big\{ -B^T \lambda^k \\ & + \beta B^T \big( Ax^{k+1} + By^{k+1} - b \big) \Big\} \geq 0, \ \forall\, y \in \mathcal{Y}. \end{aligned} \tag{2.3b}$$



Substituting $\lambda^{k+1}$ (see (2.1c)) in (2.3) (eliminating $\lambda^k$), we get $x^{k+1} \in \mathcal{X}$,

$$\begin{aligned} & \theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^T \{ -A^T \lambda^{k+1} + \beta A^T B(y^k - y^{k+1}) \\ & \qquad + (rI_{n_1} - \beta A^T A)(x^{k+1} - x^k) \} \geq 0, \ \forall x \in \mathcal{X}, \end{aligned} \tag{2.4a}$$

and

$$y^{k+1} \in \mathcal{Y}, \ \theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^T \{ -B^T \lambda^{k+1} \} \geq 0, \ \forall y \in \mathcal{Y}. \tag{2.4b}$$

For analysis convenience, we rewrite (2.4) as the following equivalent form:

$$\begin{aligned} \theta(u) - \theta(u^{k+1}) + \begin{pmatrix} x - x^{k+1} \\ y - y^{k+1} \end{pmatrix}^T & \left\{ \begin{pmatrix} -A^T \lambda^{k+1} \\ -B^T \lambda^{k+1} \end{pmatrix} + \beta \begin{pmatrix} A^T \\ B^T \end{pmatrix} B(y^k - y^{k+1}) \right. \\ & \left. + \begin{pmatrix} rI_{n_1} - \beta A^T A & 0 \\ 0 & \beta B^T B \end{pmatrix} \begin{pmatrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{pmatrix} \right\} \geq 0, \ \forall (x, y) \in \mathcal{X} \times \mathcal{Y}. \end{aligned}$$

Combining the last inequality with (2.1c), we have

$$\theta(u)-\theta(u^{k+1})+\begin{pmatrix} x-x^{k+1}\\ y-y^{k+1}\\ \lambda-\lambda^{k+1}\end{pmatrix}^T\left\{\begin{pmatrix} -A^T\lambda^{k+1}\\ -B^T\lambda^{k+1}\\ Ax^{k+1}+By^{k+1}-b\end{pmatrix}+\beta\begin{pmatrix} A^T\\ B^T\\ 0\end{pmatrix}B(y^k-y^{k+1})\right.$$

$$\left.+\begin{pmatrix} rI_{n_1}-\beta A^TA & 0 & 0\\ 0 & \beta B^TB & 0\\ 0 & 0 & \frac{1}{\beta}I_m\end{pmatrix}\begin{pmatrix} x^{k+1}-x^k\\ y^{k+1}-y^k\\ \lambda^{k+1}-\lambda^k\end{pmatrix}\right\}\ge 0,$$

for all $(x,y,\lambda)\in\mathcal{X}\times\mathcal{Y}\times\Re^m$. The above inequality can be rewritten as

$$\theta(u)-\theta(u^{k+1})+(w-w^{k+1})^TF(w^{k+1})+\beta\begin{pmatrix} x-x^{k+1}\\ y-y^{k+1}\\ \lambda-\lambda^{k+1}\end{pmatrix}^T\begin{pmatrix} A^T\\ B^T\\ 0\end{pmatrix}B(y^k-y^{k+1})$$

$$\ge\begin{pmatrix} x-x^{k+1}\\ y-y^{k+1}\\ \lambda-\lambda^{k+1}\end{pmatrix}^T\begin{pmatrix} rI_{n_1}-\beta A^TA & 0 & 0\\ 0 & \beta B^TB & 0\\ 0 & 0 & \frac{1}{\beta}I_m\end{pmatrix}\begin{pmatrix} x^k-x^{k+1}\\ y^k-y^{k+1}\\ \lambda^k-\lambda^{k+1}\end{pmatrix},\ \forall w\in\Omega. \quad (2.5)$$



## 2.2 Convergence of Linearized ADM

Based on the analysis in the last section, we have the following lemma.

**Lemma 2.1** *Let $w^{k+1}=(x^{k+1},y^{k+1},\lambda^{k+1})\in\Omega$ be generated by* (2.1) *from the given $w^k=(x^k,y^k,\lambda^k)$. Then, we have*

$$(w^{k+1}-w^*)^TG(w^k-w^{k+1})\ge(w^{k+1}-w^*)^T\eta(y^k,y^{k+1}),\ \forall\, w^*\in\Omega^*, \quad (2.6)$$

*where*

$$\eta(y^k,y^{k+1})=\beta\begin{pmatrix} A^T\\ B^T\\ 0\end{pmatrix}B(y^k-y^{k+1}), \quad (2.7)$$

*and*

$$G=\begin{pmatrix} rI_{n_1}-\beta A^TA & 0 & 0\\ 0 & \beta B^TB & 0\\ 0 & 0 & \frac{1}{\beta}I_m\end{pmatrix}. \quad (2.8)$$

**Proof**. Setting $w = w^*$ in (2.5), and using $G$ and $\eta(y^k, y^{k+1})$, we get

$$\begin{aligned}&(w^{k+1} - w^*)^T G(w^k - w^{k+1}) \\ &\quad \geq \; (w^{k+1} - w^*)^T \eta(y^k, y^{k+1}) + \theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^T F(w^{k+1}).\end{aligned}$$

Since $F$ is monotone, it follows that

$$\begin{aligned}&\theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^T F(w^{k+1}) \\ &\quad \geq \; \theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^T F(w^*) \geq 0.\end{aligned}$$

The last inequality is due to $w^{k+1} \in \Omega$ and $w^* \in \Omega^*$ is a solution of (see (1.3)). The lemma is proved. $\Box$

By using $\eta(y^k, y^{k+1})$ (see (2.7)), $Ax^* + By^* = b$ and (2.1c), we have

$$\begin{aligned}&(w^{k+1} - w^*)^T \eta(y^k, y^{k+1}) \\ &\quad = \; (B(y^k - y^{k+1}))^T \beta\{(Ax^{k+1} + By^{k+1}) - (Ax^* + By^*)\} \\ &\quad = \; (B(y^k - y^{k+1}))^T \beta(Ax^{k+1} + By^{k+1} - b) \\ &\quad = \; (\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}). \qquad (2.9)\end{aligned}$$



Substituting it in (2.6), we obtain

$$\begin{aligned}&(w^{k+1} - w^*)^T G(w^k - w^{k+1}) \\ &\quad \geq \; (\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}), \; \forall\, w^* \in \Omega^*. \qquad (2.10)\end{aligned}$$

**Lemma 2.2** *Let* $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1}) \in \Omega$ *be generated by* (2.1) *from the given* $w^k = (x^k, y^k, \lambda^k)$. *Then, we have*

$$(\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}) \geq 0. \qquad (2.11)$$

**Proof**. Since (2.4b) is true for the $k$-th iteration and the previous iteration, we have

$$\theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^T\{-B^T\lambda^{k+1}\} \geq 0, \; \forall\, y \in \mathcal{Y}, \qquad (2.12)$$

and

$$\theta_2(y) - \theta_2(y^k) + (y - y^k)^T\{-B^T\lambda^k\} \geq 0, \; \forall\, y \in \mathcal{Y}, \qquad (2.13)$$

Setting $y = y^k$ in (2.12) and $y = y^{k+1}$ in (2.13), respectively, and then adding the two resulting inequalities, we get

$$(\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}) \geq 0.$$

The assertion of this lemma is proved. $\square$

Under the assumption (2.2), the matrix $G$ is positive semi-definite. In addition, if $B$ is a full column rank matrix, $G$ is positive definite. Even if in the positive semi-definite case, we also use $\|w - \tilde{w}\|_G$ to denote

$$\|w - \tilde{w}\|_G = \sqrt{(w - \tilde{w})^T G(w - \tilde{w})}.$$

If $B$ is a full column rank matrix, $G$ is positive definite.

**Lemma 2.3** *Let $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1}) \in \Omega$ be generated by* (2.1) *from the given $w^k = (x^k, y^k, \lambda^k)$. Then, we have*

$$(w^{k+1} - w^*)^T G(w^k - w^{k+1}) \geq 0, \quad \forall\, w^* \in \Omega^*, \qquad (2.14)$$



*and consequently*

$$\|w^{k+1} - w^*\|_G^2 \leq \|w^k - w^*\|_G^2 - \|w^k - w^{k+1}\|_G^2, \; \forall\, w^* \in \Omega^*. \quad (2.15)$$

**Proof**. Assertion (2.14) follows from (2.10) and (2.11) directly. By using (2.14), we have

$$\begin{aligned}
\|w^k - w^*\|_G^2 &= \|(w^{k+1} - w^*) + (w^k - w^{k+1}\|_G^2 \\
&= \|w^{k+1} - w^*\|_G^2 + 2(w^{k+1} - w^*)^T G(w^k - w^{k+1}) \\
&\quad + \|w^k - w^{k+1}\|_G^2 \\
&\geq \|w^{k+1} - w^*\|_G^2 + \|w^k - w^{k+1}\|_G^2,
\end{aligned}$$

and thus (2.15) is proved. $\square$

The inequality (2.15) is essential for the convergence of the alternating direction method. Note that $G$ is positive semi-definite.

$$\|w^k - w^{k+1}\|_G^2 = 0 \quad \Longleftrightarrow \quad G(w^k - w^{k+1}) = 0.$$

The inequality (2.15) can be written as

$$\begin{aligned}
&\|x^{k+1} - x^*\|^2_{(rI-\beta A^T A)} + \beta\|B(y^{k+1} - y^*)\|^2 + \frac{1}{\beta}\|\lambda^{k+1} - \lambda^*\|^2 \\
&\le \quad \|x^k - x^*\|^2_{(rI-\beta A^T A)} + \beta\|B(y^k - y^*)\|^2 + \frac{1}{\beta}\|\lambda^k - \lambda^*\|^2 \\
&\qquad -\big(\|x^k - x^{k+1}\|^2_{(rI-\beta A^T A)} + \beta\|B(y^k - y^{k+1})\|^2 + \frac{1}{\beta}\|\lambda^k - \lambda^{k+1}\|^2\big).
\end{aligned}$$

It leads to that

$$\lim_{k\to\infty} x^k = x^*, \qquad \lim_{k\to\infty} By^k = By^* \qquad \text{and} \qquad \lim_{k\to\infty} \lambda^k = \lambda^*.$$

The linearizing ADM is also known as the split inexact Uzawa method in image processing literature [9, 10].



# 3 Self-Adaptive ADM-based Contraction Method

In the last section, we get $x^{k+1}$ by solving the following $x$-subproblem:

$$\min\big\{\theta_1(x) + \beta x^T A^T(Ax^k + By^k - b - \tfrac{1}{\beta}\lambda^k) + \frac{r}{2}\|x - x^k\|^2 \mid x \in \mathcal{X}\big\}$$

and it required that the parameter $r$ to satisfy

$$rI_n - \beta A^T A \succeq 0 \qquad \Longleftrightarrow \qquad r > \beta\lambda_{\max}(A^T A).$$

In some practical problem, a conservative estimation of $\lambda_{\max}(A^T A)$ will leads a slow convergence. In this section, based on the linearized ADM, we consider the self-adaptive contraction methods. Each iteration of the self-adaptive contraction methods consists of two steps–prediction step and correction step. From the given $w^k$, the prediction step produces a test vector $\tilde{w}^k$ and the correction step offers the new iterate $w^{k+1}$.

## 3.1 Prediction

1. First, for given $(x^k, y^k, \lambda^k)$, $\tilde{x}^k$ is the solution of the following problem

$$\tilde{x}^k = \text{Argmin}\left( \begin{array}{c} \big\{\theta_1(x) + \beta x^T A^T (Ax^k + By^k - b - \frac{1}{\beta}\lambda^k) \\ + \frac{r}{2}\|x - x^k\|^2 \quad \big| \; x \in \mathcal{X}\big\} \end{array} \right) \tag{3.1a}$$

2. Then, use $\lambda^k$ and the obtained $\tilde{x}^k$, $\tilde{y}^k$ is the solution of the problem

$$\tilde{y}^k = \text{Argmin}\left\{ \begin{array}{c} \theta_2(y) - (\lambda^k)^T(A\tilde{x}^k + By - b) \\ + \frac{\beta}{2}\|A\tilde{x}^k + By - b\|^2 \end{array} \middle| \, y \in \mathcal{Y} \right\} \tag{3.1b}$$

3. Finally,

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + B\tilde{y}^k - b). \tag{3.1c}$$

The subproblems in (3.1) are similar as in (2.1). Instead of $(x^{k+1}, y^{k+1}, \lambda^{k+1})$ in (2.1), we denote the output of (3.1) by $(\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)$.



**Requirements on parameters $\beta, r$** For given $\beta > 0$, choose $r$ such that

$$\beta\|A^T A(x^k - \tilde{x}^k)\| \le \nu r \|x^k - \tilde{x}^k\|, \quad \nu \in (0, 1). \tag{3.2}$$

If $rI - \beta A^T A \succ 0$, then (3.2) is satisfied. Thus, (2.2) is sufficient for (3.2).

**Analysis of the optimal conditions of subproblems in (3.1)**

Because we get $\tilde{w}^k$ in (3.1) via substituting $w^{k+1}$ in (2.1) by $\tilde{w}^k$. Therefore, similar as (2.5), we get

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ -B^T\tilde{\lambda}^k \\ A\tilde{x}^k + B\tilde{y}^k - b \end{pmatrix} + \beta \begin{pmatrix} A^T \\ B^T \\ 0 \end{pmatrix} B(y^k - \tilde{y}^k) \right.$$
$$\left. + \begin{pmatrix} rI_{n_1} - \beta A^T A & 0 & 0 \\ 0 & \beta B^T B & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} \tilde{x}^k - x^k \\ \tilde{y}^k - y^k \\ \tilde{\lambda}^k - \lambda^k \end{pmatrix} \right\} \ge 0, \;\forall w \in \Omega.$$

The last variational inequality can be rewritten as

$$\begin{array}{l}\tilde{w}^k \in \Omega, \quad \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T \\ \qquad \big\{F(\tilde{w}^k) + \eta(y^k, \tilde{y}^k) + HM(\tilde{w}^k - w^k)\big\} \geq 0, \ \forall\, w \in \Omega, \end{array} \tag{3.3}$$

where

$$\eta(y^k, \tilde{y}^k) = \beta \begin{pmatrix} A^T \\ B^T \\ 0 \end{pmatrix} B(y^k - \tilde{y}^k), \tag{3.4}$$

$$H = \begin{pmatrix} rI_{n_1} & 0 & 0 \\ 0 & \beta B^T B & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix}, \tag{3.5}$$



and

$$M = \begin{pmatrix} I_{n_1} - \frac{\beta}{r} A^T A & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & 0 & I_m \end{pmatrix}. \tag{3.6}$$

Based on the above analysis, we have the following lemma.

**Lemma 3.1** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (3.1) *from the given* $w^k = (x^k, y^k, \lambda^k)$. *Then, we have*

$$(\tilde{w}^k - w^*)^T HM(w^k - \tilde{w}^k) \geq (\tilde{w}^k - w^*)^T \eta(y^k, \tilde{y}^k), \quad \forall\, w^* \in \Omega^*. \tag{3.7}$$

**Proof**. Setting $w = w^*$ in (3.3), we obtain

$$\begin{aligned} & (\tilde{w}^k - w^*)^T HM(w^k - \tilde{w}^k) \\ \geq \quad & (\tilde{w}^k - w^*)^T \eta(y^k, \tilde{y}^k) \\ & +\theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k). \end{aligned} \tag{3.8}$$

Since $F$ is monotone and $\tilde{w}^k \in \Omega$, it follows that

$$\begin{aligned} &\theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k) \\ &\quad \geq \quad \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(w^*) \geq 0. \end{aligned}$$

The last inequality is due to $\tilde{w}^k \in \Omega$ and $w^* \in \Omega^*$ is a solution of (1.3). Substituting it in the right hand side of (3.8), the lemma is proved. $\square$

In addition, because $Ax^* + By^* = b$ and $\beta(A\tilde{x}^k + B\tilde{y}^k - b) = \lambda^k - \tilde{\lambda}^k$, we have

$$\begin{aligned} &(\tilde{w}^k - w^*)^T \eta(y^k, \tilde{y}^k) \\ &\quad = \quad (B(y^k - \tilde{y}^k))^T \beta\{(A\tilde{x}^k + B\tilde{y}^k) - (Ax^* + By^*)\} \\ &\quad = \quad (\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k). \end{aligned} \tag{3.9}$$

**Lemma 3.2** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (3.1) *from the given* $w^k = (x^k, y^k, \lambda^k)$. *Then, we have*

$$(w^k - w^*)^T HM(w^k - \tilde{w}^k) \geq \varphi(w^k, \tilde{w}^k), \quad \forall\, w^* \in \Omega^*, \tag{3.10}$$



*where*

$$\varphi(w^k, \tilde{w}^k) = (w^k - \tilde{w}^k)^T HM(w^k - \tilde{w}^k) + (\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k). \tag{3.11}$$

**Proof**. From (3.7) and (3.9) we have

$$(\tilde{w}^k - w^*)^T HM(w^k - \tilde{w}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k).$$

Assertion (3.10) follows from the last inequality and the definition of $\varphi(w^k, \tilde{w}^k)$ directly. $\square$

## 3.2 The Primary Contraction Methods

The primary contraction methods use $M(w^k - \tilde{w}^k)$ as search direction and the unit step length. In other words, the new iterate is given by

$$w^{k+1} = w^k - M(w^k - \tilde{w}^k). \tag{3.12}$$

According to (3.6), it can be written as

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} \tilde{x}^k + \frac{\beta}{r} A^T A(x^k - \tilde{x}^k) \\ \tilde{y}^k \\ \tilde{\lambda}^k \end{pmatrix}. \tag{3.13}$$

In the primary contraction method, only the $x$-part of the corrector is different from the predictor. In the method of Section 2, we need $r \geq \beta\|A^T A\|$. By using the method in this section, we need only a $r$ to satisfy the condition (3.2). In practical computation, we try to use the average of the eigenvalues of $\beta A^T A$.

Using (3.10), we have

$$\begin{aligned} &\|w^k - w^*\|_H^2 - \|w^{k+1} - w^*\|_H^2 \\ &\quad = \quad \|w^k - w^*\|_H^2 - \|(w^k - w^*) - M(w^k - \tilde{w}^k)\|_H^2 \\ &\quad = \quad 2(w^k - w^*)^T HM(w^k - \tilde{w}^k) - \|M(w^k - \tilde{w}^k)\|_H^2 \\ &\quad \geq \quad 2\varphi(w^k, \tilde{w}^k) - \|M(w^k - \tilde{w}^k)\|_H^2. \end{aligned} \tag{3.14}$$



Because $(\tilde{y}^k, \tilde{\lambda}^k) = (y^{k+1}, \lambda^{k+1})$, the inequality (2.11) is still holds and thus

$$(\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k) \geq 0.$$

Therefore, it follows from (3.14) , (3.11) and the last inequality that

$$\begin{aligned} &\|w^k - w^*\|_H^2 - \|w^{k+1} - w^*\|_H^2 \\ &\quad \geq \quad 2(w^k - \tilde{w}^k)^T HM(w^k - \tilde{w}^k) - \|M(w^k - \tilde{w}^k)\|_H^2. \end{aligned} \tag{3.15}$$

**Lemma 3.3** *Under the condition* (3.2)*, we have*

$$\begin{aligned} &2(w^k - \tilde{w}^k)^T HM(w^k - \tilde{w}^k) - \|M(w^k - \tilde{w}^k)\|_H^2 \\ &\geq (1 - \nu^2) r\|x^k - \tilde{x}^k\|^2 + \beta\|B(y^k - \tilde{y}^k)\|^2 + \tfrac{1}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2. \end{aligned} \tag{3.16}$$

**Proof**. First, we have

$$\begin{aligned} &2(w^k - \tilde{w}^k)^T HM(w^k - \tilde{w}^k) - \|M(w^k - \tilde{w}^k)\|_H^2 \\ &\quad = \quad (w^k - \tilde{w}^k)^T \big(M^T H + HM - M^T HM\big)(w^k - \tilde{w}^k). \end{aligned}$$

By using the structure of the matrices $H$ and $M$ (see (3.5) and (3.6)), we obtain

$$M^T H + HM - M^T HM = H - (I - M^T)H(I - M)$$
$$= \begin{pmatrix} rI_n & 0 & 0 \\ 0 & \beta B^T B & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix} - \begin{pmatrix} r(\frac{\beta}{r} A^T A)^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Therefore,

$$\begin{aligned} & 2(w^k - \tilde{w}^k)^T HM(w^k - \tilde{w}^k) - \|M(w^k - \tilde{w}^k)\|_H^2 \\ = \quad & \|w^k - \tilde{w}^k\|_H^2 - r\big(\tfrac{\beta^2}{r^2}\big)\|A^T A(x^k - \tilde{x}^k)\|^2. \end{aligned} \tag{3.17}$$

Under the condition (3.2), we have

$$\big(\tfrac{\beta^2}{r^2}\big)\|A^T A(x^k - \tilde{x}^k)\|^2 \le \nu^2 \|x^k - \tilde{x}^k\|^2.$$

Substituting it in (3.17), the assertion of this lemma is proved. $\square$

**Theorem 3.1** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (3.1) *from the given*



$w^k = (x^k, y^k, \lambda^k)$ *and the new iterate* $w^{k+1}$ *is given by* (3.12)*. The sequence* $\{w^k = (x^k, y^k, \lambda^k)\}$ *generated by the elementary contraction method satisfies*

$$\|w^{k+1} - w^*\|_H^2 \le \|w^k - w^*\|_H^2 - (1 - \nu^2)\|w^k - \tilde{w}^k\|_H^2. \tag{3.18}$$

**Proof**. From (3.15) and (3.16) we obtain

$$\begin{aligned} & \|w^k - w^*\|_H^2 - \|w^{k+1} - w^*\|_H^2 \\ \ge \quad & (1 - \nu^2) r\|x^k - \tilde{x}^k\|^2 + \beta\|B(y^k - \tilde{y}^k)\|^2 + \tfrac{1}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2 \\ \ge \quad & (1 - \nu^2)\|w^k - \tilde{w}^k\|_H^2. \end{aligned}$$

The assertion of this theorem is proved. $\square$

Theorem 3.1 is essential for the convergence of the primary contraction method.

## 3.3 The general contraction method

**The general contraction method**

For given $w^k$, we use

$$w(\alpha) = w^k - \alpha M(w^k - \tilde{w}^k) \tag{3.19}$$

to update the $\alpha$-dependent new iterate. For any $w^* \in \Omega^*$, we define

$$\vartheta(\alpha) := \|w^k - w^*\|_H^2 - \|w(\alpha) - w^*\|_H^2 \tag{3.20}$$

and

$$q(\alpha) = 2\alpha\varphi(w^k, \tilde{w}^k) - \alpha^2\|M(w^k - \tilde{w}^k)\|_H^2, \tag{3.21}$$

where $\varphi(w^k, \tilde{w}^k)$ is defined in (3.11).

**Theorem 3.2** *Let $w(\alpha)$ be defined by (3.19). For any $w^* \in \Omega^*$ and $\alpha \geq 0$, we have*

$$\vartheta(\alpha) \geq q(\alpha), \tag{3.22}$$

*where $\vartheta(\alpha)$ and $q(\alpha)$ are defined in* (3.20) *and* (3.21)*, respectively.*



**Proof**. It follows from (3.19) and (3.20) that

$$\begin{aligned}
\vartheta(\alpha) &= \|w^k - w^*\|_H^2 - \|(w^k - w^*) - \alpha M(w^k - \tilde{w}^k)\|_H^2 \\
&= 2\alpha(w^k - w^*)^T HM(w^k - \tilde{w}^k) - \alpha^2\|M(w^k - \tilde{w}^k)\|_H^2.
\end{aligned}$$

By using (3.10) and the definition of $q(\alpha)$, the theorem is proved. $\square$

Note that $q(\alpha)$ in (3.21) is a quadratic function of $\alpha$ and it reaches its maximum at

$$\alpha^* = \frac{\varphi(w^k, \tilde{w}^k)}{\|M(w^k - \tilde{w}^k)\|_H^2}. \tag{3.23}$$

In practical computation, we use

$$w^{k+1} = w^k - \gamma\alpha_k^* M(w^k - \tilde{w}^k), \tag{3.24}$$

to update the new iterate, where $\gamma \in [1, 2)$ is a relaxation factor. By using (3.20) and (3.22), we have

$$\|w^{k+1} - w^*\|_H^2 \leq \|w^k - w^*\|_H^2 - q(\gamma\alpha_k^*). \tag{3.25}$$

Note that

$$q(\gamma\alpha_k^*) = 2\gamma\alpha_k^*\varphi(w^k, \tilde{w}^k) - (\gamma\alpha_k^*)^2\|M(w^k - \tilde{w}^k)\|_H^2. \tag{3.26}$$

Using (3.23) and (3.24), we obtain

$$q(\gamma\alpha_k^*) = \gamma(2-\gamma)(\alpha_k^*)^2\|M(w^k - \tilde{w}^k)\|_H^2 = \frac{2-\gamma}{\gamma}\|w^k - w^{k+1}\|_H^2,$$

and consequently it follows from (3.25) that

$$\|w^{k+1} - w^*\|_H^2 \le \|w^k - w^*\|_H^2 - \frac{2-\gamma}{\gamma}\|w^k - w^{k+1}\|_H^2, \ \forall\, w^* \in \Omega^*. \tag{3.27}$$

On the other hand, it follows from (3.23) and (3.26) that

$$q(\gamma\alpha_k^*) = \gamma(2-\gamma)\alpha_k^*\varphi(w^k, \tilde{w}^k). \tag{3.28}$$



By using (3.11) and (3.16), we obtain

$$\begin{aligned}
&2\varphi(w^k, \tilde{w}^k) - \|M(w^k - \tilde{w}^k)\|_H^2 \\
&\quad = \ 2(w^k - \tilde{w}^k)^T HM(w^k - \tilde{w}^k) - \|M(w^k - \tilde{w}^k)\|_H^2 \\
&\qquad\quad +2(\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k) \\
&\quad \ge (1-\nu^2)r\|x^k - \tilde{x}^k\|^2 + \beta\|B(y^k - \tilde{y}^k)\|^2 + \tfrac{1}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2 \\
&\qquad\quad +2(\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k) \\
&\quad = \ (1-\nu^2)r\|x^k - \tilde{x}^k\|^2 + \beta\|B(y^k - \tilde{y}^k) + \tfrac{1}{\beta}(\lambda^k - \tilde{\lambda}^k)\|^2.
\end{aligned}$$

Thus, we have $2\varphi(w^k, \tilde{w}^k) > \|M(w^k - \tilde{w}^k)\|_H^2$ and consequently

$$\alpha_k^* > \frac{1}{2}.$$

In addition, because

$$
\begin{aligned}
\varphi(w^k, \tilde{w}^k) &= (w^k - \tilde{w}^k)^T HM(w^k - \tilde{w}^k) + (\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k) \\
&= \left(\begin{array}{c} x^k - \tilde{x}^k \\ y^k - \tilde{y}^k \\ \lambda^k - \tilde{\lambda}^k \end{array}\right)^T \left(\begin{array}{ccc} rI_{n_1} - \beta A^T A & 0 & 0 \\ 0 & \beta B^T B & \frac{1}{2} B^T \\ 0 & \frac{1}{2} B & \frac{1}{\beta} I_m \end{array}\right) \left(\begin{array}{c} x^k - \tilde{x}^k \\ y^k - \tilde{y}^k \\ \lambda^k - \tilde{\lambda}^k \end{array}\right) \\
&\geq \|x^k - \tilde{x}^k\| \cdot \big(r\|x^k - \tilde{x}^k\| - \beta \|A^T A(x^k - \tilde{x}^k)\|\big) \\
&\quad + \frac{1}{2} \left(\begin{array}{c} y^k - \tilde{y}^k \\ \lambda^k - \tilde{\lambda}^k \end{array}\right)^T \left(\begin{array}{cc} \beta B^T B & 0 \\ 0 & \frac{1}{\beta} I_m \end{array}\right) \left(\begin{array}{c} y^k - \tilde{y}^k \\ \lambda^k - \tilde{\lambda}^k \end{array}\right).
\end{aligned}
$$

Under the condition (3.2), it follows from the last inequality that

$$
\begin{aligned}
\varphi(w^k, \tilde{w}^k) &\geq \min\big\{(1-\nu), \frac{1}{2}\big\} \|w^k - \tilde{w}^k\|_H^2 \\
&\geq \frac{1-\nu}{2} \|w^k - \tilde{w}^k\|_H^2. \qquad (3.29)
\end{aligned}
$$



By using (3.25), (3.28), (3.29) and $\alpha_k^* \geq \frac{1}{2}$, we obtain the following theorem for the general contraction method.

**Theorem 3.3** *The sequence* $\{w^k = (x^k, y^k, \lambda^k)\}$ *generated by the general contraction method satisfies*

$$
\begin{aligned}
&\|w^{k+1} - w^*\|_H^2 \\
&\quad \leq \|w^k - w^*\|_H^2 - \tfrac{\gamma(2-\gamma)(1-\nu)}{4} \|w^k - \tilde{w}^k\|_H^2, \ \forall w^* \in \Omega^*. \qquad (3.30)
\end{aligned}
$$

The inequality (3.30) in Theorem 3.3 is essential for the convergence of the general contraction method.

Both the inequalities (3.18) and (3.30) can be written as

$$
\|w^{k+1} - w^*\|_H^2 \leq \|w^k - w^*\|_H^2 - c_0 \|w^k - \tilde{w}^k\|_H^2, \ \forall w^* \in \Omega^*,
$$

where $c_0 > 0$ is a constant. Therefore, we have

$$
\begin{aligned}
& r\|x^{k+1} - x^*\|^2 + \beta\|B(y^{k+1} - y^*)\|^2 + \frac{1}{\beta}\|\lambda^{k+1} - \lambda^*\|^2 \\
\le \quad & r\|x^k - x^*\|^2 + \beta\|B(y^k - y^*)\|^2 + \frac{1}{\beta}\|\lambda^k - \lambda^*\|^2 \\
& -c_0\big(r\|x^k - \tilde{x}^k\|^2 + \beta\|B(y^k - \tilde{y}^k)\|^2 + \frac{1}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2\big).
\end{aligned}
$$

It leads to that

$$
\lim_{k\to\infty}\big(r\|x^k - \tilde{x}^k\|^2 + \beta\|B(y^k - \tilde{y}^k)\|^2 + \frac{1}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2\big),
$$

and

$$
\lim_{k\to\infty} x^k = x^*, \qquad \lim_{k\to\infty} By^k = By^* \qquad \text{and} \qquad \lim_{k\to\infty} \lambda^k = \lambda^*.
$$



# 4 Applications in $l_1$-norm problems

An important $l_1$-norm problem in the area machine learning is the $l_1$ regularized linear regression, also called the lasso [8]. This involves solving

$$
\min \ \tau\|x\|_1 + \tfrac{1}{2}\|Ax - b\|_2^2, \tag{4.1}
$$

where $\tau > 0$ is a scalar regularization parameter that is usually chosen by cross-validation. In typical applications, there are many more features than training examples, and the goal is to find a parsimonious model for the data. The problem (1.1) can be reformulated to problem

$$
\begin{array}{ll}
\min & \tau\|x\|_1 + \frac{1}{2}\|y\|_2^2 \\
 & Ax - y = b
\end{array} \tag{4.2}
$$

which is a form of (1.1). Applied the alternating direction method (1.4) to the problem (4.2), the $x$-subproblem is

$$
x^{k+1} = \text{Argmin}\,\{\tau\|x\|_1 + \tfrac{\beta}{2}\|(Ax - y^k) - \tfrac{1}{\beta}\lambda^k\|_2^2\},
$$

and the solution does not has closed form. Applied the linearized alternating direction method (2.1) to the problem (4.2), the $x$-subproblem (2.1a) is

$$\tilde{x}^k = \text{Argmin}\{\tau\|x\|_1 + \frac{r}{2}\|x - [x^k + \frac{1}{r}\lambda^k - \frac{\beta}{r}A^T(Ax^k - y^k)]\|^2\}. \quad (4.3)$$

This problem is form of (1.5) and its solution has the following closed form:

$$\tilde{x}^k = a - P_{B_\infty^{\tau/r}}[a], \qquad \text{where} \qquad a = x^k + \frac{1}{r}\lambda^k - \frac{\beta}{r}A^T(Ax^k - y^k)$$

and

$$B_\infty^{\tau/r} = \{\xi \in \Re^n \,|\, -(\tau/r)e \le \xi \le (\tau/r)e\}.$$

By using the linearized alternating direction method in Section 2, for given $\beta > 0$, it needs $r > \beta\lambda_{\max}(A^TA)$. By using the self-adaptive ADM-based contraction method in Section 3, it needs $r$ to satisfy

$$\beta\|A^TA(x^k - \tilde{x}^k)\| \le \nu r\|x^k - \tilde{x}^k\|, \quad \nu \in (0,1).$$

Because $A$ is a generic matrix, the above condition is satisfied even if $r$ is much less than $\beta\lambda_{\max}(A^TA)$.



# References


[1] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Foundations and Trends in Machine Learning, **3**, 1-122, 2010.

[2] D. Gabay, Applications of the Method of Multipliers to Variational Inequalities, in *Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems*, M. Fortin and R. Glowinski, eds., North Holland, Amsterdam, The Netherlands, 299-331, 1983.

[3] D. Gabay and B. Mercier, A dual algorithm for the solution of nonlinear variational problems viafinite element approximations, Computers and Mathematics with Applications, 2:17-40, 1976.

[4] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[5] B.S. He, A class of projection and contraction methods for monotone variational inequalities, Applied Math. $\&$ Optim., **35**, 69-76, 1997.

[6] B.S. He, Inexact implicit methods for monotone general variational inequalities, Math. Program. Series A, **86**, 199-217, 1999.

[7] B.S. He, L-Z Liao, D.R. Han and H. Yang, A new inexact alternating directions method for monotone variational inequalities, Math. Progr., **92**, 103-118, 2002.

[8] R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B, **58**, 267 – 288, 1996.

[9] X. Q. Zhang, M. Burger, X. Bresson and S. Osher, Bregmanized nonlocal regularization for deconvolution and sparse reconstruction, SIAM J. Imag. Sci., **3**, 253-276, 2010.

[10] X. Q. Zhang, M. Burger and S. Osher, A unified primal-dual algorithm framework based on Bregman iteration, J. Sci. Computing, **46**, 20-46, 2010.

# 凸优化和单调变分不等式的收缩算法

## 第十三讲: 定制 PPA 算法意义下的乘子交替方向法

Alternating direction method
in sense of customized PPA

南京大学数学系　　何炳生
hebma@nju.edu.cn

The context of this lecture is based on the manuscript [2]



## 1 Structured constrained convex optimization

We consider the following structured constrained convex optimization problem

$$\min \{\theta_1(x)+\theta_2(y) \mid Ax+By=b,\ x\in\mathcal{X},\ y\in\mathcal{Y}\} \tag{1.1}$$

where $\theta_1(x):\Re^{n_1}\to\Re,\ \theta_2(y):\Re^{n_2}\to\Re$ are convex functions (but not necessary smooth), $A\in\Re^{m\times n_1}$, $B\in\Re^{m\times n_2}$ and $b\in\Re^m$, $\mathcal{X}\subset\Re^{n_1}$, $\mathcal{Y}\subset\Re^{n_2}$ are given closed convex sets.

The task of solving the problem (1.1) is to find an $(x^*,y^*,\lambda^*)\in\Omega$, such that

$$\left\{\begin{array}{l}\theta_1(x)-\theta_1(x^*)+(x-x^*)^T(-A^T\lambda^*)\ge 0,\\ \theta_2(y)-\theta_2(y^*)+(y-y^*)^T(-B^T\lambda^*)\ge 0, \quad \forall\,(x,y,\lambda)\in\Omega,\\ (\lambda-\lambda^*)^T(Ax^*+By^*-b)\ge 0,\end{array}\right. \tag{1.2}$$

where

$$\Omega=\mathcal{X}\times\mathcal{Y}\times\Re^m.$$

By denoting

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

and

$$\theta(u) = \theta_1(x) + \theta_2(y),$$

the first order optimal condition (1.2) can be written in a compact form such as

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \geq 0, \ \forall \, w \in \Omega. \quad (1.3)$$

Note that the mapping $F$ is monotone. We use $\Omega^*$ to denote the solution set of the variational inequality (1.3). For convenience we use the notations

$$v = \begin{pmatrix} y \\ \lambda \end{pmatrix} \quad \text{and} \quad \mathcal{V}^* = \{(y^*, \lambda^*) \,|\, (x^*, y^*, \lambda^*) \in \Omega^*\}.$$



## Applied ADMM to the structure VI $(y^k, \lambda^k) \Rightarrow (y^{k+1}, \lambda^{k+1})$

First, for given $(y^k, \lambda^k)$, $\tilde{x}^k$ is the solution of the following problem

$$\tilde{x}^k = \text{Argmin}\left\{ \begin{array}{c} \theta_1(x) - (\lambda^k)^T(Ax + By^k - b) \\ + \frac{\beta}{2}\|Ax + By^k - b\|^2 \end{array} \middle| \, x \in \mathcal{X} \right\} \quad (1.4a)$$

Use $\lambda^k$ and the obtained $\tilde{x}^k$, $\tilde{y}^k$ is the solution of the following problem

$$\tilde{y}^k = \text{Argmin}\left\{ \begin{array}{c} \theta_2(y) - (\lambda^k)^T(A\tilde{x}^k + By - b) \\ + \frac{\beta}{2}\|A\tilde{x}^k + By - b\|^2 \end{array} \middle| \, y \in \mathcal{Y} \right\} \quad (1.4b)$$

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + B\tilde{y}^k - b). \quad (1.4c)$$

The sub-problems (1.4a) and (1.4b) are separately solved.

**Classical Alternating Direction Method of Multipliers:**

$$v^{k+1} = \tilde{v}^k.$$

**Ye-Yuan's Alternating Direction Method of Multipliers:**

$$v^{k+1} = v^k - \alpha_k(v^k - \tilde{v}^k), \qquad \alpha_k = \gamma\alpha_k^*, \qquad \gamma \in (0,2) \tag{1.5a}$$

where

$$\alpha_k^* = \frac{\|v^k - \tilde{v}^k\|_H^2 + (\lambda^k - \tilde{\lambda}^k)^T B(y^k - \tilde{y}^k)}{\|v^k - \tilde{v}^k\|_H^2} \tag{1.5b}$$

and

$$\|v^k - \tilde{v}^k\|_H^2 = \beta\|B(y^k - \tilde{y}^k)\|^2 + \frac{1}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2.$$

The convergence of the classical alternating direction method and Ye-Yuan's ADMM are demonstrated in Lecture 11.



**Ye-Yuan's ADMM vs Classical ADMM:**

The iteration number of Ye-Yuan's ADMM is less the one of the classical ADMM. However, in Ye-Yuan's ADMM, we need to calculate the step size length $\alpha_k^*$ in each iteration.

## 2 ADMM based customized PPA

The $k$-th iteration of the proposed Alternating Direction Method of Multipliers in this section is also from a pair of $(y^k, \lambda^k)$ to a new pair of $(y^{k+1}, \lambda^{k+1})$. In the prediction step, we generate a $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)$ which satisfies

$$\tilde{w}^k \in \Omega, \;\; \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T\{F(\tilde{w}^k) + Q(\tilde{v}^k - v^k)\} \ge 0, \;\; \forall\, w \in \Omega, \tag{2.1}$$

where $Q$ is a $3 \times 2$ block matrix whose first low is zero, and the rest sub-matrix is symmetric and positive semi-definite. In details, the matrices $Q$ and $M$ have the following forms

$$Q = \begin{pmatrix} 0 & 0 \\ \beta B^T B & -B^T \\ -B & \frac{1}{\beta} I_m \end{pmatrix} \quad \text{and} \quad H = \begin{pmatrix} \beta B^T B & -B^T \\ -B & \frac{1}{\beta} I_m \end{pmatrix}. \tag{2.2}$$

Note that the matrix $H$ is symmetric and positive semidefinite. If we replace $Q(\tilde{v}^k - v^k)$ by $G(\tilde{w}^k - w^k)$ with a symmetric positive definite matrix $G$, then (2.1) becomes a sub-problem of the proximal point algorithm. Thus, the method in this lecture is called the ADMM-based customized PPA or Alternating direction method in the sense of customized PPA.

## 2.1 Motivation

In the classical ADMM, the variable $x$ is not a part of the state. $\tilde{x}^k$ is only an intermediate result computed from the previous state $(y^k, \lambda^k)$. Note that $\tilde{x}^k$ is



the minimizer of the augmented Lagrangian function with $y = y^k$, *i.e.,*

$$\tilde{x}^k = \text{Argmin}\{\theta_1(x) - (\lambda^k)^T(Ax + By^k - b) + \tfrac{\beta}{2}\|Ax + By^k - b\|^2 \mid x \in \mathcal{X}\}. \tag{2.3}$$

Thus, we have $\tilde{x}^k \in \mathcal{X}$ and

$$\theta_1(x) - \theta_1(\tilde{x}^k) + (x - \tilde{x}^k)^T\{-A^T\lambda^k + \beta A^T\big(A\tilde{x}^k + By^k - b\big)\} \ge 0, \ \forall\, x \in \mathcal{X}. \tag{2.4}$$

If we write the above variational inequality as

$$\theta_1(x) - \theta_1(\tilde{x}^k) + (x - \tilde{x}^k)^T\{-A^T\tilde{\lambda}^k\} \ge 0, \ \forall\, x \in \mathcal{X},$$

it implies that

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k - b). \tag{2.5}$$

According to the above definition, for any $\tilde{y}^k \in \mathcal{Y}$, we have

$$(A\tilde{x}^k + B\tilde{y}^k - b) - B(\tilde{y}^k - y^k) + \tfrac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) = 0. \tag{2.6}$$

Combining (2.4) and (2.6) together, we get $(\tilde{x}^k, \tilde{\lambda}^k) \in \mathcal{X} \times \Re^m$,

$$\theta_1(x) - \theta_1(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k + B\tilde{y}^k - b \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ -B & \frac{1}{\beta} \end{pmatrix} \begin{pmatrix} \tilde{y}^k - y^k \\ \tilde{\lambda}^k - \lambda^k \end{pmatrix} \right\} \geq 0, \quad (2.7)$$

for all $(x, \lambda) \in \mathcal{X} \times \Re^m$. In order to get $\tilde{w}^k \in \Omega$, such that

$$\begin{pmatrix} \big(\theta_1(x) - \theta_1(\tilde{x}^k)\big) + \\ \big(\theta_2(y) - \theta_2(\tilde{y}^k)\big) \end{pmatrix} + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ -B^T\tilde{\lambda}^k \\ A\tilde{x}^k + B\tilde{y}^k - b \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \beta B^T B & -B^T \\ -B & \frac{1}{\beta} \end{pmatrix} \begin{pmatrix} \tilde{y}^k - y^k \\ \tilde{\lambda}^k - \lambda^k \end{pmatrix} \right\} \geq 0, \ \forall w \in \Omega, \quad (2.8)$$



we need only to find $\tilde{y}^k \in \mathcal{Y}$, such that

$$\tilde{y}^k \in \mathcal{Y}, \quad \big(\theta_2(y) - \theta_2(\tilde{y}^k)\big) + (y - \tilde{y}^k)^T \left\{ -B^T\tilde{\lambda}^k + B^T\big(\beta B(\tilde{y}^k - y^k) - (\tilde{\lambda}^k - \lambda^k)\big) \right\} \geq 0, \ \forall\, y \in \mathcal{Y}. \quad (2.9)$$

By using (2.5), we have

$$\beta B(\tilde{y}^k - y^k) - (\tilde{\lambda}^k - \lambda^k) = \beta\big(A\tilde{x}^k + B\tilde{y}^k - b\big).$$

Thus, the variational inequality (2.9) is

$$\big(\theta_2(y) - \theta_2(\tilde{y}^k)\big) + (y - \tilde{y}^k)^T \left\{ -B^T\tilde{\lambda}^k + \beta B^T\big(A\tilde{x}^k + B\tilde{y}^k - b\big) \right\} \geq 0, \ \forall\, y \in \mathcal{Y}.$$

For given $\tilde{x}^k$ and the defined $\tilde{\lambda}^k$ in (2.5), such a $\tilde{y}^k$ can be obtained via solving the following convex optimization problem:

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + \tfrac{\beta}{2}\|A\tilde{x}^k + By - b - \tfrac{1}{\beta}\tilde{\lambda}^k\|^2 \mid y \in \mathcal{Y}\}. \quad (2.10)$$

The above analysis guides us to construct the ADMM based customized PPA.

## 2.2 The proposed ADMM based customized PPA

From given $v^k = (y^k, \lambda^k)$, the prediction step produces $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)$.

The prediction step:

1. First, for given $(y^k, \lambda^k)$, $\tilde{x}^k$ is the solution of the following problem

$$\tilde{x}^k = \text{Argmin}\{\theta_1(x) + \tfrac{\beta}{2}\|Ax + By^k - b - \tfrac{1}{\beta}\lambda^k\|^2 \,|\, x \in \mathcal{X}\} \quad (2.11a)$$

2. Set the multipliers by

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k - b). \quad (2.11b)$$

3. Finally, use the obtained $\tilde{x}^k$ and $\tilde{\lambda}^k$, find $\tilde{y}^k$ by

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + \tfrac{\beta}{2}\|A\tilde{x}^k + By - b - \tfrac{1}{\beta}\tilde{\lambda}^k\|^2 \,|\, y \in \mathcal{Y}\} \quad (2.11c)$$

In the ADMM view of point, we generate the predictor in the order

$$\tilde{x}^k, \quad \tilde{\lambda}^k \quad \text{and} \quad \tilde{y}^k.$$



As illustrated in the motivation, we get (2.8). This variational inequality can be written in the form of

$$\tilde{w}^k \in \Omega, \ (w - \tilde{w}^k)^T\{F(\tilde{w}^k) + Q(\tilde{v}^k - v^k)\} \geq 0, \ \forall\, w \in \Omega, \quad (2.12)$$

where $Q$ is just the same matrix defined in (2.2). The above variational inequality is essential in the unified framework of the contraction methods.

The correction step: Update the new iterate $v^{k+1}$ by

$$v^{k+1} = v^k - \gamma(v^k - \tilde{v}^k), \quad \gamma \in (0, 2). \quad (2.13)$$

To get the new iterate $v^{k+1}$, this method does not need to calculate the step size.

## 2.3 Convergence of the ADMM in sense of customized PPA

Based on the analysis in the last subsection, we have the following lemma.

**Lemma 2.1** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (2.11) *from the given*

$v^k = (y^k, \lambda^k)$. *Then, we have*

$$(\tilde{w}^k - w^*)^T Q(v^k - \tilde{v}^k) \geq 0, \ \forall w^* \in \Omega^*, \qquad (2.14)$$

*where the matrix $Q$ is defined in* (2.2).

**Proof**. Setting $(x, y, \lambda) = (x^*, y^*, \lambda^*)$ in (2.8), we obtain

$$(\tilde{w}^k - w^*)^T Q(v^k - \tilde{v}^k) \geq \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k). \quad (2.15)$$

Since $F$ is monotone and $\tilde{w}^k \in \Omega$, it follows that

$$\begin{aligned} & \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k) \\ \geq \quad & \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(w^*) \geq 0. \end{aligned}$$

The last inequality is due to $\tilde{w}^k \in \Omega$ and $w^* \in \Omega^*$ (see (1.3)). Therefore, the right hand side of (2.15) is non-negative and the lemma is proved. $\Box$

**Lemma 2.2** *Let $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ be generated by* (2.11) *from the given*



$v^k = (y^k, \lambda^k)$. *Then, we have*

$$(v^k - v^*)^T H(v^k - \tilde{v}^k) \geq \|v^k - \tilde{v}^k\|_H^2, \ \forall \, v^* \in \mathcal{V}^*, \qquad (2.16)$$

*where $M$ is defined in* (2.2).

**Proof**. Recall the matrices $Q$ and $H$ in (2.2). It follows from (2.14) that

$$(\tilde{v}^k - v^*)^T H(v^k - \tilde{v}^k) \geq 0, \ \forall \, v^* \in \mathcal{V}^*.$$

Assertion (2.16) follows from the last inequality directly. $\Box$

The matrix $H$ is symmetric and positive semi-definite. We still use $\|v - \tilde{v}\|_H$ to denote that

$$\|v - \tilde{v}\|_H = \sqrt{(v - \tilde{v})^T H(v - \tilde{v})}.$$

If $\|v^k - \tilde{v}^k\|_H^2 = 0$, because $H$ is symmetric and positive semi-definite, we have $H(v^k - \tilde{v}^k) = 0$. In this case, $\tilde{w}^k$ is a solution of the variational inequality (see (1.2) and (2.8)). Thus, we can take $\|v^k - \tilde{v}^k\|_H^2 \leq \epsilon$ as the stopping criterium in the iteration process.

**Theorem 2.1** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (2.11) *from the given* $v^k = (y^k, \lambda^k)$ *and the new iterate* $v^{k+1}$ *be given by* (2.13). *Then we have*

$$\|v^{k+1} - v^*\|_H^2 \le \|v^k - v^*\|_H^2 - \gamma(2-\gamma)\|v^k - \tilde{v}^k\|_H^2, \ \forall\, v^* \in \mathcal{V}^*. \quad (2.17)$$

**Proof**. By a simple manipulation, we obtain

$$\begin{aligned}
&\|v^{k+1} - v^*\|_H^2 \\
&\overset{(2.16)}{=} \quad \|(v^k - v^*) - \gamma(v^k - \tilde{v}^k)\|_H^2 \\
&= \quad \|v^k - v^*\|_H^2 - 2\gamma(v^k - v^*)^T H(v^k - \tilde{v}^k) + \gamma^2\|v^k - \tilde{v}^k\|_H^2 \\
&\overset{(2.13)}{\le} \quad \|v^k - v^*\|_H^2 - 2\gamma\|v^k - \tilde{v}^k\|_H^2 + \gamma^2\|v^k - \tilde{v}^k\|_H^2 \\
&= \quad \|v^k - v^*\|_H^2 - \gamma(2-\gamma)\|v^k - \tilde{v}^k\|_H^2.
\end{aligned}$$

This is true for any $v^* \in \mathcal{V}^*$ and the theorem is proved. $\square$

The inequality (2.17) is essential for the convergence of the proposed alternating direction method. The detailed convergence proof can be found in [2]. For the convergence rate of the customized PPA, the reader are refereed to [9].



## 2.4 Ensure the matrix $H$ to be positive definite

In the ADMM based customized PPA (2.11), the subproblem (2.11c) can be written as

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + \tfrac{\beta}{2}\|By - p^k\|^2 \,|\, y \in \mathcal{Y}\}, \quad (2.18)$$

where

$$p^k = b + \tfrac{1}{\beta}\tilde{\lambda}^k - A\tilde{x}^k.$$

If we add an additional term $\frac{\delta\beta}{2}\|B(y - y^k)\|^2$ (with any small $\delta > 0$) to the objective function of the subproblem (2.11c), we will get $\tilde{y}^k$ via

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + \tfrac{\beta}{2}\|By - p^k\|^2 + \tfrac{\delta\beta}{2}\|B(y - y^k)\|^2 \,|\, y \in \mathcal{Y}\}.$$

By a manipulation, the solution point of the above subproblem is obtained via

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + \tfrac{(1+\delta)\beta}{2}\|By - q^k\|^2 \,|\, y \in \mathcal{Y}\}, \quad (2.19)$$

where

$$q^k = \tfrac{1}{1+\delta}(p^k + \delta y^k).$$

In this way, the matrix $Q$ in (2.12) will be modified to

$$Q = \begin{pmatrix} 0 & 0 \\ (1+\delta)\beta B^T B & -B^T \\ -B & \frac{1}{\beta} I_m \end{pmatrix},$$

and the related matrix $H$ in (2.2) becomes

$$\begin{aligned} H &= \begin{pmatrix} (1+\delta)\beta B^T B & -B^T \\ -B & \frac{1}{\beta} I_m \end{pmatrix} \\ &= \begin{pmatrix} \sqrt{\beta} B^T & 0 \\ 0 & \sqrt{\frac{1}{\beta}} I_m \end{pmatrix} \begin{pmatrix} (1+\delta) I & -I \\ -I & I_m \end{pmatrix} \begin{pmatrix} \sqrt{\beta} B & 0 \\ 0 & \sqrt{\frac{1}{\beta}} I_m \end{pmatrix}. \end{aligned} \quad (2.20)$$

Thus, for any $\delta > 0$, $H$ is positive definite when $B$ is a full rank matrix. In other words, instead of (2.18), using (2.19) to get $\tilde{y}^k$, it will ensure the positivity of $H$ theoretically. However, in practical computation, it works still well by using $\delta = 0$.



# 3 Application and Numerical Experiments

## 3.1 Applications to least-squares problems

We consider the following problem:

$$\min\{\frac{1}{2}\|X - C\|_F^2 \,|\, X \in S_+^n \cap S_B\}, \quad (3.1)$$

where

$$S_+^n = \{H \in R^{n\times n} \,|\, H^T = H, \; H \succeq 0\}.$$

and

$$S_B = \{H \in R^{n\times n} \,|\, H^T = H, \; H_L \le H \le H_U\}.$$

**Use the following MATLAB Code to produce the matrices $C$, $H_L$ and $H_U$**

```
rand('state',0);   C=rand(n,n);      C=(C'+C)-ones(n,n)+eye(n);
%% C is symmetric and C_{ij} is in (-1,1), C_{jj} is in (0,2)
HU=ones(n)*0.1;   HL=-HU;  for i=1:n HU(i,i)=1; HL(i,i)=1; end;
```

**The problem is converted to the following equivalent one:**

$$\begin{array}{rl} \min & \frac{1}{2}\|X-C\|^2 + \frac{1}{2}\|Y-C\|^2 \\ \text{s.t} & X - Y = 0, \\ & X \in S_+^n,\ Y \in S_B. \end{array} \tag{3.2}$$

**The basic sub-problems in the ADMM based customized PPA**

- For fixed $Y^k$ and $Z^k$,

$$\tilde{X}^k = \text{Argmin}\{\frac{1}{2}\|X-C\|_F^2 - \mathbf{Tr}(Z^kX) + \frac{\beta}{2}\|X-Y^k\|_F^2 \mid X \in S_+^n\}$$

- Set $\tilde{Z}^k$ by

$$\tilde{Z}^k = Z^k - \beta(\tilde{X}^k - Y^k).$$

- With fixed $\tilde{X}^k$ and $\tilde{Z}^k$,

$$\tilde{Y}^k = \text{Argmin}\{\frac{1}{2}\|Y-C\|_F^2 + \mathbf{Tr}(\tilde{Z}^kY) + \frac{\beta}{2}\|\tilde{X}^k - Y\|_F^2 \mid Y \in S_B\}$$



**$\tilde{X}^k$ can be directly obtained via**

$$\tilde{X}^k = P_{S_+^n}\{\frac{1}{1+\beta}(\beta Y^k + Z^k + C)\}. \tag{3.3}$$

$$P_{S_+^n}(A) = U\Lambda^+U^T, \qquad [U,\Lambda] = \text{eig}(A), \qquad \Lambda^+ = \max(\Lambda, 0).$$

**Similarly, $\tilde{Y}^k$ in is given by**

$$\tilde{Y}^k = P_{S_B}\{\frac{1}{1+\beta}(\beta\tilde{X}^k - \tilde{Z}^k + C)\}. \tag{3.4}$$

$$S_B = \{H \mid H_L \le H \le H_U\}, \qquad P_{S_B}(A) = \min(\max(H_L, A), H_U)$$

**The most time consuming calculation is $[U,\Lambda] = \textbf{eig}(A)$, $9n^3$**

## MATLAB Code – An iteration of the classical ADMM

```
Y0= Y;                          Z0     = Z;             k = k+1;
X = (Y0*beta+Z0+C)/(1+beta);    [V,D] = eig(X);         D = max(0,D);
X = (V*D)*V';
Y = min(max((X*beta-Z0+C)/(1+beta),HL),HU);
Z = Z0-(X-Y)*beta;
```

## MATLAB Code – An iteration of the new order ADMM

```
Y0= Y;                          Z0     = Z;             k = k+1;
X = (Y0*beta+Z0+C)/(1+beta);    [V,D] = eig(X);         D = max(0,D);
X = (V*D)*V';                   Z      = Z0-(X-Y0)*beta;;
Y = min(max((X*beta-Z+C)/(1+beta),HL),HU);
```

## MATLAB Code – An iteration of the extended ADMM

```
Y0= Y;                          Z0     = Z;             k = k+1;
X = (Y0*beta+Z0+C)/(1+beta);    [V,D] = eig(X);         D = max(0,D);
X = (V*D)*V';                   Z      = Z0-(X-Y0)*beta;;
Y = min(max((X*beta-Z+C)/(1+beta),HL),HU);
Y = Y0-(Y0-Y)*1.5;
Z = Z0-(Z0-Z)*1.5;
```



### Numerical results for problem (3.1)

C=rand(n,n); C=(C'+C)-ones(n,n) + eye(n)

$H_U$=ones(n,n)/10; $H_L$=-ones(n,n)/10; $H_U(jj) = H_L(jj)$=1.

Table 1. Numerical results

| $n \times n$ Matrix | Classical ADMM | | Customized PPA | | Extended C-PPA | | |
|---|---|---|---|---|---|---|---|
| $n =$ | No. It | CPU Sec. | No. It | CPU Sec. | No. It | CPU Sec. | $\beta$ |
| 100 | 46 | 1.39 | 44 | 1.37 | 28 | 0.94 | 5 |
| 200 | 50 | 3.07 | 50 | 3.05 | 31 | 4.41 | 10 |
| 500 | 48 | 25.50 | 49 | 24.52 | 32 | 16.50 | 10 |
| 800 | 51 | 110.18 | 50 | 107.29 | 33 | 72.12 | 10 |
| 1000 | 51 | 208.93 | 52 | 212.74 | 34 | 140.70 | 10 |
| 2000 | 55 | 1578.96 | 55 | 1579.68 | 36 | 1053.87 | 10 |

## 3.2 Applications to image restoration

The mathematical form of the image restoration problem is

$$\min \||\nabla x|\|_1 + \frac{\mu}{2}\|Kx - f\|^2, \tag{3.5}$$

where $\mu > 0$ is trade-off; $K$ is a blur operator and $f$ is observed image.

**The equivalent problem:**

$$\begin{aligned} &\min \||y|\|_1 + \frac{\mu}{2}\|Kx - f\|^2 \\ &\text{s. t. } \nabla x = y, \end{aligned} \tag{3.6}$$

This is a problem of form (1.1) where $\mathcal{X}$, $\mathcal{Y}$ are full spaces,

$$\begin{aligned} &\theta_1(x) = \frac{\mu}{2}\|Kx - f\|^2, \\ &\theta_2(y) = \||y|\|_1, \\ &A = \nabla, \quad B = -I \quad \text{and} \quad b = 0. \end{aligned}$$



**The augmented Lagrangian function**

$$\mathcal{L}_A(x, y, \lambda) = \||y|\|_1 + \frac{\mu}{2}\|Kx - f\|^2 - \lambda^T(\nabla x - y) + \frac{\beta}{2}\|\nabla x - y\|^2,$$

where $\lambda$ is Lagrange multiplier and $\beta$ is the penalty parameter.

**For given $(y^k, \lambda^k)$, get $(\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)$ as follows:**

1. $\tilde{x}^k$ is the solution of the following least square problem

$$\tilde{x}^k = \arg\min_x \left\{ \tfrac{\mu}{2}\|Kx - f\|^2 - (\lambda^k)^T(\nabla x - y^k) + \tfrac{\beta}{2}\|\nabla x - y^k\|^2 \right\}.$$

2. Set $\tilde{\lambda}^k$ by

$$\tilde{\lambda}^k = \lambda^k - \beta(\nabla \tilde{x}^k - y^k).$$

3. Finally, with fixed $(\tilde{x}^k, \tilde{\lambda}^k)$, $\tilde{y}^k$ are solutions of

$$\tilde{y}^k = \arg\min_y \left\{ \||y|\|_1 - (\tilde{\lambda}^k)^T(\nabla \tilde{x}^k - y) + \tfrac{\beta}{2}\|\nabla \tilde{x}^k - y\|^2 \right\}.$$

### Solving the $x$ subproblem for getting $\tilde{x}^k$:

$$(\beta\nabla^T\nabla + \mu K^T K)\tilde{x}^k = \nabla^T(\beta y^k + \lambda^k) + \mu K^T f.$$

- If $\nabla$ and $K$ satisfy some periodic boundary conditions, they can be factored by Fourier transform as $\nabla = \mathcal{F}^{-1}\Lambda_D\mathcal{F}$ and $K = \mathcal{F}^{-1}\Lambda_K\mathcal{F}$.
- If $\nabla$ and $K$ satisfy some reflective boundary conditions, they can be factored by discrete cosine transform as $\nabla = \mathcal{C}^{-1}\Lambda_D\mathcal{C}$ and $K = \mathcal{C}^{-1}\Lambda_K\mathcal{C}$.

### Solving the $y$ subproblem for getting $\tilde{y}^k$:

$$\tilde{y}^k = \text{shrink}_{\frac{1}{\beta}}\left(\nabla\tilde{x}^k - \frac{\tilde{\lambda}^k}{\beta}\right),$$

where

$$\text{shrink}_c(v) = v - \min(c, \|v\|)\frac{v}{\|v\|}.$$



Note that

$$\text{shrink}_c(v) = v - P_{B_2^c}(v) \quad \text{where} \quad B_2^c = \{v \in \Re^n \,|\, \|v\|_2 \le c\}.$$

## MATLAB Code – An iteration of the classical ADMM

```
%%%%%%%%%%%%%%%%%%%%   step 1   x^{k+1}   %%%%%%%%%%%%%%%%%%%%%%
Temp= PTx(beta*v1+lbd11) + PTy(beta*v2+lbd12) + HTx0;
un   = real(ifft2(fft2(Temp)./MDu));
%%%%%%%%%%%%%%%%%%%%   step 2   y^{k+1}   %%%%%%%%%%%%%%%%%%%%%%
dxun= Px(un);
dyun= Py(un);
sk1  = dxun - lbd11/beta;
sk2  = dyun - lbd12/beta;
nsk  = sqrt(sk1.^2 + sk2.^2); nsk(nsk==0)=1;
nsk  = max(1-1./(beta*nsk),0);
vn1  = sk1.*nsk;
vn2  = sk2.*nsk;
%%%%%%%%%%%%%%%%%%%% update  \lambda     %%%%%%%%%%%%%%%%%%%%%%
lbdn11 = lbd11 - beta*(dxun - vn1);
lbdn12 = lbd12 - beta*(dyun - vn2);
%%%%%%%%%%%%%%%%%%%% New iterative point %%%%%%%%%%%%%%%%%%%%%%
u = un; v1 = vn1; v2 = vn2; lbd11 = lbdn11; lbd12 = lbdn12;
```

## MATLAB Code – An iteration of the new order ADMM

```
%%%%%%%%%%%%%%%%%%%%   step 1  x^{k+1}  %%%%%%%%%%%%%%%%%%%%%%
Temp= PTx(beta*v1+lbd11) + PTy(beta*v2+lbd12) + HTx0;
un  = real(ifft2(fft2(Temp)./MDu));
dxun= Px(un);
dyun= Py(un);
%%%%%%%%%%%%%%%%%%%%   update \lambda   %%%%%%%%%%%%%%%%%%%%%%
lbdn11 = lbd11 - beta*(dxun - v1);
lbdn12 = lbd12 - beta*(dyun - v2);
%%%%%%%%%%%%%%%%%%%%   step 2  y^{k+1}  %%%%%%%%%%%%%%%%%%%%%%
sk1 = dxun - lbdn11/beta;
sk2 = dyun - lbdn12/beta;
nsk = sqrt(sk1.^2 + sk2.^2); nsk(nsk==0)=1;
nsk = max(1-1./(beta*nsk),0);
vn1 = sk1.*nsk;
vn2 = sk2.*nsk;
%%%%%%%%%%%%%%%%%%%% New iterative point %%%%%%%%%%%%%%%%%%%%%%
u = un; v1 = vn1; v2 = vn2; lbd11 = lbdn11; lbd12 = lbdn12;
```



## MATLAB Code – An iteration of the extended C-PPA

```
%%%%%%%%%%%%%%%%%%%%   step 1  x^{k+1}  %%%%%%%%%%%%%%%%%%%%%%
Temp= PTx(beta*v1+lbd11) + PTy(beta*v2+lbd12) + HTx0;
un  = real(ifft2(fft2(Temp)./MDu));
dxun= Px(un);
dyun= Py(un);
%%%%%%%%%%%%%%%%%%%%  update \lambda    %%%%%%%%%%%%%%%%%%%%%%
lbdn11 = lbd11 - beta*(dxun - v1);
lbdn12 = lbd12 - beta*(dyun - v2);
%%%%%%%%%%%%%%%%%%%%  step 2  y^{k+1}  %%%%%%%%%%%%%%%%%%%%%%
sk1 = dxun - lbd11/beta;
sk2 = dyun - lbd12/beta;
nsk = sqrt(sk1.^2 + sk2.^2); nsk(nsk==0)=1;
nsk = max(1-1./(beta*nsk),0);
vn1 = sk1.*nsk;
vn2 = sk2.*nsk;
%%%%%%%%%%%%%%%%%% New iterative point %%%%%%%%%%%%%%%%%%%%%
u  = un;
v1 = v1 - gamma*(v1-vn1);
v2 = v2 - gamma*(v2-vn2);
lbd11 = lbd11 - gamma*(lbd11-lbdn11);
lbd12 = lbd12 - gamma*(lbd12-lbdn12);
```

## Numerical results for image restoration

```
I = double(imread('chart.tiff'))/255; I = double(imread('house.png'))/255;
h = fspecial('disk',7); x0 = imfilter(I,h,'circular')+0.02*randn(size(I));
```

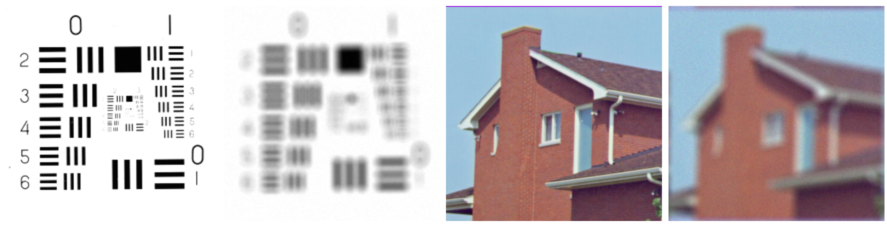


Figure 1: Original and degraded images. Left: Chart. Right: House



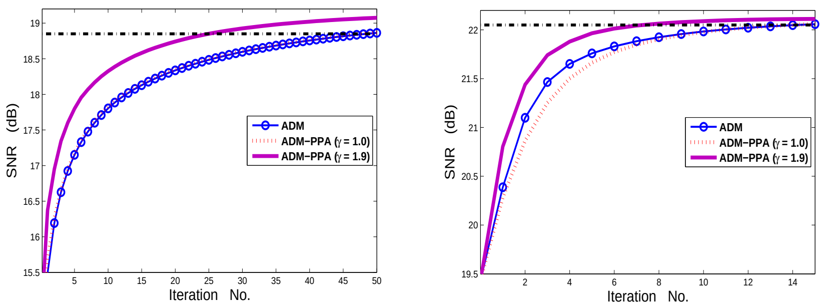



Figure 2: Performances of ADMM and two variants methods on TV-l2. Left: Chart. Right: House.

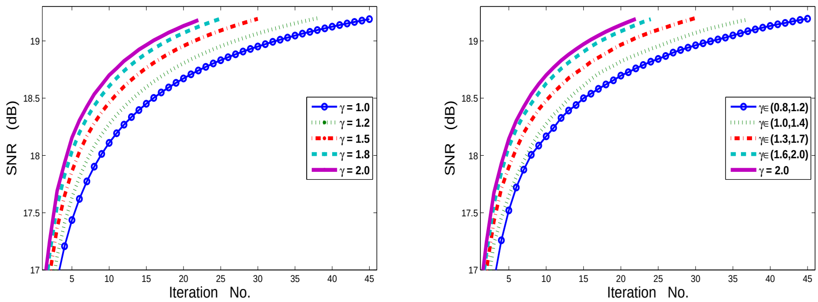



Figure 3: Performances of Algorithm 2 with different values of $\gamma$ for Chart. Top: fixed $\gamma$. Bottom: random generated $\gamma$.



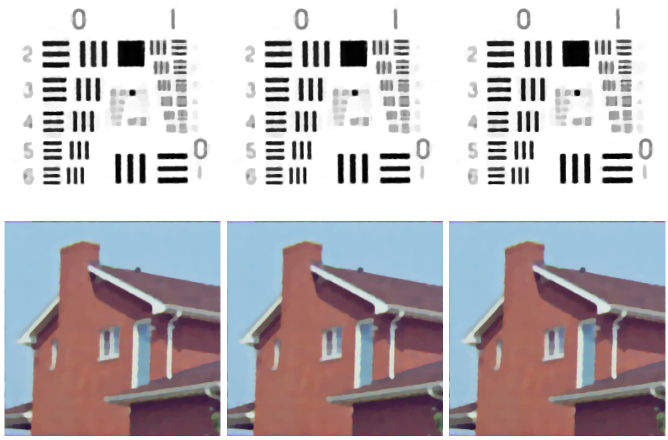


Figure 4: Restorations. From left column to right column: ADMM, new order ADMM, and the extended new order ADMM.

Table 1: Numerical comparisons of the classical ADMM (ADMM), the customized PPA and the extended customized PPA for TV-$l^2$ image restoration.

| | Chart | | | House | | |
|---|---|---|---|---|---|---|
| | ADMM | C-PPA | E-C-PPA | ADMM | C-PPA | E-C-PPA |
| | | $\gamma = 1$ | $\gamma = 1.8$ | | $\gamma = 1$ | $\gamma = 1.8$ |
| Iter | 74 | 74 | 42 | 57 | 56 | 33 |
| CPU | 2.32 | 2.29 | 1.29 | 2.79 | 2.75 | 1.63 |
| SNR | 19.01 | 19.01 | 19.02 | 22.11 | 22.11 | 22.12 |

SNR $= 20\log_{10}\frac{\|x\|}{\|x-I\|}$, where $x$ is restoration and $I$ is original image.

It seems that the new order ADMM is so good as the classical one. However, the extended-ADMM converges much faster than the other both ADMMs.

**Remark** For solving the structured convex optimization problem (1.1), the



classical alternating direction method is described in (1.4) and then the new iterate is updated by $(y^{k+1},\lambda^{k+1}) = (\tilde{y}^k,\tilde{\lambda}^k)$.

In [7], it was shown that the ADMM is the application of the Douglas-Rachford splitting method [13] to the dual of (1.1). Then, in [4], Eckstein and Bertsekas demonstrated that the Douglas-Rachford splitting method is a special form of the proximal point algorithm (PPA) in [14], and inspired by the relaxed PPA in [8], they proposed the generalized alternating direction method of multipliers

$$\left\{\begin{array}{l} x^{k+1} = \text{argmin}\big\{\theta_1(x) - x^TA^T\lambda^k + \frac{\beta}{2}\|Ax + By^k - b\|^2 \;\big|\; x\in\mathcal{X}\big\},\\ y^{k+1} = \text{argmin}\left\{\begin{array}{c}\theta_2(y) - y^TB^T\lambda^k + \\ \frac{\beta}{2}\|[\alpha Ax^{k+1} - (1-\alpha)(By^k - b)] + By - b\|^2\end{array}\middle| y\in\mathcal{Y}\right\},\\ \lambda^{k+1} = \lambda^k - \beta\{[\alpha Ax^{k+1} - (1-\alpha)(By^k - b)] + By^{k+1} - b\}, \end{array}\right. \tag{3.7}$$

where the parameter $\alpha\in(0,2)$ is a relaxation factor. The numerical efficiency of the recursion (3.7) with an over-relaxed choice of $\alpha$, especially $\alpha\in[1.5,1.8]$ empirically, has been shown in [5, 6]. Some of young researcher told me that the numerical behaviors of the customized PPA based ADMM (2.11)-(2.13) are almost the same as the relaxed ADMM (3.7). It seems possible to prove the equivalence of the two methods [3]. We emphasize here that the explanation of this lecture is in the frame of our lecture series. Use such explanation, it is easy to prove the contraction and the $\mathcal{O}(1/t)$ convergence rate of the customized PPA based ADMM and its linearized variant, for details, see the next lecture.

# References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Foundations and Trends in Machine Learning, **3**, 1-122, 2010.

[2] X.J. Cai, G.Y. Gu, B.S. He and X.M. Yuan, A relaxed customized proximal point algorithm for separable convex programming. http://www.optimization-online.org/DB_HTML/2011/08/3141.html

[3] C.H. Chen, Privat communication, Department of Mathematics, Nanjing University, 2012.

[4] J. Eckstein and D.P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Program., **55**, 293-318, 1992.

[5] J. Eckstein, Parallel alternating direction multiplier decomposition of convex programs, J. Optim. Theory Appli., **80**, 39-62, 1994.

[6] J. Eckstein and M. C. Ferris, Operator-splitting methods for monotone affine variational inequalities, with a parallel application to optimal control, INFORMS J. Comput., **10**, 218-235, 1998.



[7] D. Gabay, Applications of the Method of Multipliers to Variational Inequalities, in *Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems*, M. Fortin and R. Glowinski, eds., North Holland, Amsterdam, The Netherlands, 299-331, 1983.

[8] E. G. Gol'shtein and N. V. Tret'yakov, Modified Lagrangian in convex programming and their generalizations, Math. Program. Studies, **10**, 86-97, 1979.

[9] G.Y. Gu, B.S. He and X.M. Yuan: Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: a uniform approach. http://www.optimization-online.org/DB_HTML/2011/10/ 3220.html

[10] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[11] B.S. He, Inexact implicit methods for monotone general variational inequalities", Math. Program. Series A, **86**, 199-217, 1999

[12] B.S. He, L-Z Liao, D.R. Han and H. Yang, A new inexact alternating directions method for monotone variational inequalities, Math. Progr., **92**, 103-118, 2002

[13] P. L. Lions and B. Mercier, Splitting algorithms for the sum of two nonlinear operators, SIAM J. Num. Anal., **16**, 964-979, 1979.

[14] B. Martinet, Regularisation, d'inéquations variationelles par approximations succesives, Rev. Francaise d'Inform. Recherche Oper., **4**, 154-159, 1970.

[15] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Control Optim., **14**, 877-898, 1976.

[16] C. H. Ye and X. M. Yuan, A Descent Method for Structured Monotone Variational Inequalities, Optimization Methods and Software, **22**, 329-338, 2007.

# 凸优化和单调变分不等式的收缩算法

## 第十四讲: 定制 PPA 算法意义的线性化交替方向法

Linearized alternating direction method in sense of the customized PPA

南京大学数学系　何炳生
hebma@nju.edu.cn

The context of this lecture is based on the manuscript [2]



## 1 Structured constrained convex optimization

We consider the following structured constrained convex optimization problem

$$\boxed{\min\{\theta_1(x)+\theta_2(y)\,|\,Ax+By=b,\ x\in\mathcal{X},\ y\in\mathcal{Y}\}} \quad (1.1)$$

where $\theta_1(x):\Re^{n_1}\to\Re,\ \theta_2(y):\Re^{n_2}\to\Re$ are convex functions (but not necessary smooth), $A\in\Re^{m\times n_1}$, $B\in\Re^{m\times n_2}$ and $b\in\Re^m$, $\mathcal{X}\subset\Re^{n_1}$, $\mathcal{Y}\subset\Re^{n_2}$ are given closed convex sets.

The task of solving the problem (1.1) is to find an $(x^*,y^*,\lambda^*)\in\Omega$, such that

$$\left\{\begin{array}{l}\theta_1(x)-\theta_1(x^*)+(x-x^*)^T(-A^T\lambda^*)\ge 0,\\ \theta_2(y)-\theta_2(y^*)+(y-y^*)^T(-B^T\lambda^*)\ge 0, \quad \forall\,(x,y,\lambda)\in\Omega,\\ (\lambda-\lambda^*)^T(Ax^*+By^*-b)\ge 0,\end{array}\right. \quad (1.2)$$

where

$$\Omega=\mathcal{X}\times\mathcal{Y}\times\Re^m.$$

By denoting

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

and

$$\theta(u) = \theta_1(x) + \theta_2(y),$$

the first order optimal condition (1.2) can be written in a compact form such as

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \geq 0, \ \forall\, w \in \Omega. \quad (1.3)$$

Note that the mapping $F$ is monotone. We use $\Omega^*$ to denote the solution set of the variational inequality (1.3). For convenience we use the notations

$$v = \begin{pmatrix} y \\ \lambda \end{pmatrix} \qquad \text{and} \qquad \mathcal{V}^* = \{(y^*, \lambda^*) \,|\, (x^*, y^*, \lambda^*) \in \Omega^*\}.$$



**Applied the ADM-based customized PPA to the problem (1.1)**

From given $v^k = (y^k, \lambda^k)$, the prediction step produces $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)$.

The prediction step:

1. First, for given $(y^k, \lambda^k)$, $\tilde{x}^k$ is the solution of the following problem

$$\tilde{x}^k = \text{Argmin}\left\{ \begin{array}{c} \theta_1(x) - (\lambda^k)^T(Ax + By^k - b) \\ +\frac{\beta}{2}\|Ax + By^k - b\|^2 \end{array} \middle|\, x \in \mathcal{X} \right\} \quad (1.4a)$$

2. Set the multipliers by

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k - b). \quad (1.4b)$$

3. Finally, use the obtained $\tilde{x}^k$ and $\tilde{\lambda}^k$, find $\tilde{y}^k$ by

$$\tilde{y}^k = \text{Argmin}\left\{ \begin{array}{c} \theta_2(y) - (\tilde{\lambda}^k)^T(A\tilde{x}^k + By - b) \\ +\frac{\beta}{2}\|A\tilde{x}^k + By - b\|^2 \end{array} \middle|\, y \in \mathcal{Y} \right\} \quad (1.4c)$$

As analyzed in the last chapter, we have

$$\tilde{w}^k \in \Omega,\ \theta(u)-\theta(\tilde{u}^k)+(w-\tilde{w}^k)^T\{F(\tilde{w}^k)+Q(\tilde{v}^k-v^k)\} \geq 0,\ \forall\, w \in \Omega, \quad (1.5)$$

where

$$Q = \begin{pmatrix} 0 & 0 \\ \beta B^T B & -B^T \\ -B & \frac{1}{\beta} I_m \end{pmatrix} \quad \text{and} \quad M = \begin{pmatrix} \beta B^T B & -B^T \\ -B & \frac{1}{\beta} I_m \end{pmatrix}. \quad (1.6)$$

The new iterate $v^{k+1}$ is given by

$$v^{k+1} = v^k - \gamma(v^k - \tilde{v}^k), \quad \gamma \in (0,2).$$

The generated sequence $\{v^k\}$ satisfies

$$\|v^{k+1} - v^*\|_M^2 \leq \|v^k - v^*\|_M^2 - \|v^k - v^{k+1}\|_M^2.$$



# 2 Linearized ADM-based PPA Method

Note that the subproblems (1.4a) and (1.4c) in the last section are equivalent to the problems

$$\tilde{x}^k = \text{Argmin}\big\{\theta_1(x) + \tfrac{\beta}{2}\|(Ax + By^k - b) - \tfrac{1}{\beta}\lambda^k\|^2 \big| x \in \mathcal{X}\big\} \quad (2.1a)$$

and

$$\tilde{y}^k = \text{Argmin}\big\{\theta_2(y) + \tfrac{\beta}{2}\|(A\tilde{x}^k + By - b) - \tfrac{1}{\beta}\tilde{\lambda}^k\|^2 \big| y \in \mathcal{Y}\big\} \quad (2.1b)$$

respectively. In some structured optimization (1.1), the subproblem (2.1b) is easy because $B$ is usually a scalar matrix. However, to obtain the solution of the subproblem (2.1a) is expensive in the case that $A$ does not have a special form. In this lecture, we suppose that only the solution of the problem

$$\min\left\{\theta_1(x) + \frac{r}{2}\|x-a\|^2 \,|\, x \in \mathcal{X}\right\}$$

has a closed form, and consider to linearize the quadratic function of the subproblem (2.1a) ADM in sense of the customized PPA.

## 2.1 Linearized alternating direction method

The prediction step:

1. First, for given $(x^k, y^k, \lambda^k)$, solving the $x$ subproblem to get $\tilde{x}^k$ by

$$\tilde{x}^k = \text{Argmin}\left( \begin{array}{c} \Big\{\theta_1(x) + \beta x^T A^T(Ax^k + By^k - b - \frac{1}{\beta}\lambda^k) \\ +\frac{r}{2}\|x - x^k\|^2 \quad \Big| \, x \in \mathcal{X}\Big\} \end{array} \right). \tag{2.2a}$$

2. Set the new multipliers by

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + By^k - b). \tag{2.2b}$$

3. Finally, use the obtained $\tilde{x}^k$ and $\tilde{\lambda}^k$, solving the $y$ subproblem to get $\tilde{y}^k$ by

$$\tilde{y}^k = \text{Argmin}\left\{\theta_2(y) + \frac{\beta}{2}\|(A\tilde{x}^k + By - b) - \frac{1}{\beta}\tilde{\lambda}^k\|^2 \Big| y \in \mathcal{Y}\right\}. \tag{2.2c}$$



**Request on the parameter $r$** For given $\beta > 0$, $r$ should satisfy

$$rI - \beta A^T A \succeq 0. \tag{2.3}$$

The correction step: Update the new iterate $w^{k+1}$ by

$$w^{k+1} = w^k - \gamma(w^k - \tilde{w}^k), \quad \gamma \in [1, 2). \tag{2.4}$$

To get the new iterate $w^{k+1}$, this method does not need to calculate the step size. However, it needs to estimate the max-eigenvalue of $A^T A$, *i. e.*, $\lambda_{\max}(A^T A)$.

## 2.2 Analysis in the PPA framework

Note that the solution of (2.2a), $\tilde{x}^k$ satisfies

$$\begin{array}{l} \tilde{x}^k \in \mathcal{X}, \ \ \theta_1(x) - \theta_1(\tilde{x}^k) + (x - \tilde{x}^k)^T \\ \quad \{-A^T\lambda^k + \beta A^T(Ax^k + By^k - b) + r(\tilde{x}^k - x^k)\} \geq 0, \ \forall x \in \mathcal{X}. \end{array} \tag{2.5}$$

Substituting $\tilde{\lambda}^k$ (see (2.2b)) in (2.5) (eliminating $\lambda^k$), we get

$$\tilde{x}^k \in \mathcal{X}, \;\; \theta_1(x) - \theta_1(\tilde{x}^k) + (x - \tilde{x}^k)^T \\ \big\{-A^T\tilde{\lambda}^k + (rI - \beta A^TA)(\tilde{x}^k - x^k)\big\} \ge 0, \; \forall\, x \in \mathcal{X}. \tag{2.6}$$

The solution of (2.2c), $\tilde{y}^k$ satisfies

$$\tilde{y}^k \in \mathcal{Y}, \;\; \theta_2(y) - \theta_2(\tilde{y}^k) + (y - \tilde{y}^k)^T \\ \{-B^T\tilde{\lambda}^k + \beta B^T\big(A\tilde{x}^k + B\tilde{y}^k - b\big)\} \ge 0, \; \forall\; y \in \mathcal{Y}. \tag{2.7}$$

Note that $\beta\big(A\tilde{x}^k + B\tilde{y}^k - b\big) = (\lambda^k - \tilde{\lambda}^k) + \beta B(\tilde{y}^k - y^k)$ (see (2.2b)). Substituting it in (2.7), we obtain

$$\tilde{y}^k \in \mathcal{Y}, \;\; \theta_2(y) - \theta_2(\tilde{y}^k) + (y - \tilde{y}^k)^T \\ \big\{-B^T(2\tilde{\lambda}^k - \lambda^k) + \beta B^TB\big(\tilde{y}^k - y^k\big)\big\} \ge 0, \; \forall\; y \in \mathcal{Y}. \tag{2.8}$$

From (2.2b) we have

$$(A\tilde{x}^k + B\tilde{y}^k - b) - B(\tilde{y}^k - y^k) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) = 0. \tag{2.9}$$



Combining the inequalities (2.6), (2.8) and (2.9), we obtain

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ -B^T\tilde{\lambda}^k \\ A\tilde{x}^k + B\tilde{y}^k - b \end{pmatrix} \right. \\ \left. + \begin{pmatrix} (rI - \beta A^TA) & 0 & 0 \\ 0 & \beta B^TB & -B^T \\ 0 & -B & \frac{1}{\beta}I_m \end{pmatrix} \begin{pmatrix} \tilde{x}^k - x^k \\ \tilde{y}^k - y^k \\ \tilde{\lambda}^k - \lambda^k \end{pmatrix} \right\} \ge 0, \; \forall\, w \in \Omega. \tag{2.10}$$

The last variational inequality can be written in form of

$$\tilde{w}^k \in \Omega, \; (w - \tilde{w}^k)^T\{F(\tilde{w}^k) + G(\tilde{w}^k - w^k)\} \ge 0, \; \forall\, w \in \Omega, \tag{2.11}$$

where

$$G = \begin{pmatrix} (rI - \beta A^TA) & 0 & 0 \\ 0 & \beta B^TB & -B^T \\ 0 & -B & \frac{1}{\beta}I_m \end{pmatrix} \tag{2.12}$$

which is essential in the framework of the PPA contraction methods.

## 2.3 Convergence of the Linearized ADM-based PPA Method

Based on the analysis in the last subsection, we have the following lemma.

**Lemma 2.1** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (2.2) *from the given* $w^k = (x^k, y^k, \lambda^k)$*. Then, we have*

$$(\tilde{w}^k - w^*)^T G(w^k - \tilde{w}^k) \geq 0, \ \forall w^* \in \Omega^*, \tag{2.13}$$

*where* $G$ *is defined in* (2.12)*.*

**Proof**. Setting $w = w^*$ in (2.10), we get

$$(\tilde{w}^k - w^*)^T G(w^k - \tilde{w}^k) \geq \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k).$$

Since $F$ is monotone and $\tilde{w}^k \in \Omega$, it follows that

$$\theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k) \geq \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(w^*).$$



The right hand side of the last inequality is non-negative because $\tilde{w}^k \in \Omega$ and $w^* \in \Omega^*$. the assertion follows directly. $\square$

**Lemma 2.2** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (2.2) *from the given* $w^k = (x^k, y^k, \lambda^k)$*. Then, we have*

$$(w^k - w^*)^T G(w^k - \tilde{w}^k) \geq \|w^k - \tilde{w}^k\|_G^2, \ \forall\, w^* \in \Omega^*, \tag{2.14}$$

*where* $G$ *is defined in* (2.12)*.*

**Proof**. Assertion (2.14) follows from the last inequality directly. $\square$

Since $G$ is symmetric and positive semi-definite, we have

$$w^k - \tilde{w}^k = 0 \quad \text{or} \quad G(w^k - \tilde{w}^k) = 0,$$

whenever $\|w^k - \tilde{w}^k\|_G^2 = 0$. Therefore, it follows from (2.10) that $\tilde{w}^k$ is a solution of the variational inequality when $\|w^k - \tilde{w}^k\|_G^2 = 0$.

**Theorem 2.1** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (2.2) *from the given*

$w^k = (x^k, y^k, \lambda^k)$ *and the new iterate* $w^{k+1}$ *be given by* (2.4)*. Then we have*

$$\|w^{k+1}-w^*\|_G^2 \le \|w^k-w^*\|_G^2-\gamma(2-\gamma)\|w^k-\tilde{w}^k\|_G^2, \ \forall\, w^* \in \Omega^*. \quad (2.15)$$

**Proof**. By using (2.4) and (2.14), we obtain

$$\begin{aligned} &\|w^{k+1} - w^*\|_G^2 \\ &\overset{(2.4)}{=} \quad \|(w^k - w^*) - \gamma(w^k - \tilde{w}^k)\|_G^2 \\ &\overset{(2.14)}{\le} \quad \|w^k - w^*\|_G^2 - 2\gamma\|w^k - \tilde{w}^k\|_G^2 + \gamma^2\|w^k - \tilde{w}^k\|_G^2 \\ &= \quad \|w^k - w^*\|_G^2 - \gamma(2-\gamma)\|w^k - \tilde{w}^k\|_G^2. \end{aligned}$$

This is true for any $w^* \in \Omega^*$ and the theorem is proved. $\square$

The inequality (2.15) is essential for the convergence of the Linearized alternating direction method. By using (2.4), the result of Theorem 2.1 can be written as

$$\|w^{k+1} - w^*\|_G^2 \le \|w^k - w^*\|_G^2 - \frac{2-\gamma}{\gamma}\|w^k - w^{k+1}\|_G^2, \ \forall\, w^* \in \Omega^*.$$



# 3 Convergence rate of L-ADM-based C-PPA Method

**Lemma 3.1** *Let* $\{w^k\}$ *be the sequence generated by the customized PPA (2.2) with (2.4). Then, we have*

$$(\tilde{w}^k - \tilde{w}^{k+1})^T G\{(w^k - w^{k+1}) - (\tilde{w}^k - \tilde{w}^{k+1})\} \ge 0. \quad (3.1)$$

**Proof**. Set $w = \tilde{w}^{k+1}$ in (2.11), we have

$$\theta(\tilde{u}^{k+1}) - \theta(\tilde{u}^k) + (\tilde{w}^{k+1} - \tilde{w}^k)^T\{F(\tilde{w}^k) + G(\tilde{w}^k - w^k)\} \ge 0. \quad (3.2)$$

Note that (2.11) is also true for $k := k+1$ and thus we have

$$\theta(u)-\theta(\tilde{u}^{k+1})+(w-\tilde{w}^{k+1})^T\{F(\tilde{w}^{k+1})+G(\tilde{w}^{k+1}-w^{k+1})\} \ge 0, \ \forall\, w \in \Omega.$$

Set $w = \tilde{w}^k$ in the above inequality, we obtain

$$\theta(\tilde{u}^k) - \theta(\tilde{u}^{k+1}) + (\tilde{w}^k - \tilde{w}^{k+1})^T\{F(\tilde{w}^{k+1}) + G(\tilde{w}^{k+1} - w^{k+1})\} \ge 0. \quad (3.3)$$

Adding (3.2) and (3.3) and using the monotonicity of $F$, we get

$$(\tilde{w}^k - \tilde{w}^{k+1})^T G\{(w^k - w^{k+1}) - (\tilde{w}^k - \tilde{w}^{k+1})\} \geq 0.$$

we obtain (3.1) immediately. $\square$

**Lemma 3.2** *Let $\{w^k\}$ be the sequence generated by the customized PPA (2.2) with (2.4). Then, we have*

$$\begin{aligned}&(w^k - \tilde{w}^k)^T G\{(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\} \\ \geq\ & \frac{1}{\gamma}\|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_G^2. \end{aligned} \tag{3.4}$$

**Proof**. Adding the term $\|(w^k - w^{k+1}) - (\tilde{w}^k - \tilde{w}^{k+1})\|_G^2$ to the both sides of (3.1), we obtain

$$\begin{aligned}&(w^k - w^{k+1})^T G\{(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\} \\ \geq\ & \|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_G^2. \end{aligned} \tag{3.5}$$



Substituting the term $(w^k - w^{k+1})$ in the left hand side of (3.5) by $\gamma(w^k - \tilde{w}^k)$ (see (2.4)), we obtain (3.4) and the lemma is proved. $\square$

**Lemma 3.3** *Let $\{w^k\}$ be the sequence generated by the customized PPA (2.2) with (2.4). Then, we have*

$$\|w^{k+1} - \tilde{w}^{k+1}\|_G^2 \leq \|w^k - \tilde{w}^k\|_G^2. \tag{3.6}$$

**Proof**. Setting $a = w^k - \tilde{w}^k$ and $b = w^{k+1} - \tilde{w}^{k+1}$ in the identity

$$\|a\|_G^2 - \|b\|_G^2 = 2a^T G(a-b) - \|a-b\|_G^2,$$

we obtain

$$\begin{aligned}&\|w^k - \tilde{w}^k\|_G^2 - \|w^{k+1} - \tilde{w}^{k+1}\|_G^2 \\ =\ & 2(w^k - \tilde{w}^k)^T G\{(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\} \\ & -\|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_G^2. \end{aligned}$$

By using (3.4) to the first term of the right hand side of the last equality, we obtain

$$\|w^k-\tilde{w}^k\|_G^2-\|w^{k+1}-\tilde{w}^{k+1}\|_G^2 \geq \frac{2-\gamma}{\gamma}\|(w^k-\tilde{w}^k)-(w^{k+1}-\tilde{w}^{k+1})\|_G^2.$$

The assertion of this lemma is proved. $\Box$

Having the assertion (2.15) and Lemma 3.3, we are ready to present the $O(1/t)$ convergence rate of the customized PPA in the residue sense.

**Theorem 3.1** *Let $\{w^k\}$ be the sequence generated by the customized PPA (2.2) with (2.4). Then, we have*

$$\|w^k-\tilde{w}^k\|_G^2 \leq \frac{1}{(k+1)\gamma(2-\gamma)}\|w^0-w^*\|_G^2, \quad \forall w^* \in \Omega^*. \tag{3.7}$$

**Proof**. First, it follows from (2.15) that

$$\gamma(2-\gamma)\sum_{t=0}^{\infty}\|w^t-\tilde{w}^t\|_G^2 \leq \|w^0-w^*\|_G^2, \quad \forall\, w^* \in \Omega^*. \tag{3.8}$$



According to Lemma 3.3, the sequence $\{\|w^t-\tilde{w}^t\|_H^2\}$ is non-increasing. Therefore, we have

$$(k+1)\|w^k-\tilde{w}^k\|_G^2 \leq \sum_{i=0}^{k}\|w^i-\tilde{w}^i\|_G^2. \tag{3.9}$$

The assertion of this theorem follows from (3.8) and (3.9) directly. $\Box$

The solution set of the variational inequality VI$(\Omega, F, \theta)$ is convex and closed. Theorem 3.1 indicates that ADMM has $O(1/k)$ iteration convergence rate. Let

$$d = \inf\{\|w^0-w^*\|_G \,|\, w^* \in \Omega^*\}.$$

For any given $\epsilon > 0$, in order to enforce the error $\|w^k-\tilde{w}^k\|_G^2 \leq \epsilon$, according to (3.7), it needs at most $k = \lfloor d^2/\gamma(2-\gamma)\epsilon \rfloor$ iterations.

# 4 ADM-based Contraction Method

In ADM-based contraction methods, we use the $\tilde{w}^k$ generated by (2.2) to construct a search direction.

## 4.1 Contraction Method

The prediction step of the contraction method in this subsection is the same as (2.2). Therefore, we have (2.10) and rewrite it as

$$\tilde{w}^k \in \Omega, \;\; \theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ -B^T\tilde{\lambda}^k \\ A\tilde{x}^k + B\tilde{y}^k - b \end{pmatrix} \right.$$

$$\left. + \begin{pmatrix} rI_n & 0 & 0 \\ 0 & \beta B^T B & -B^T \\ 0 & -B & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} (I_n - \frac{\beta}{r} A^T A)(\tilde{x}^k - x^k) \\ \tilde{y}^k - y^k \\ \tilde{\lambda}^k - \lambda^k \end{pmatrix} \right\} \geq 0, \;\; \forall w \in \Omega.$$



Again, the above variational inequality can be written in form of

$$\begin{aligned} &\tilde{w}^k \in \Omega, \; \theta(u) - \theta(\tilde{u}^k) \\ &\quad + (w - \tilde{w}^k)^T \{F(\tilde{w}^k) + Q(\tilde{w}^k - w^k)\} \geq 0, \; \forall\, w \in \Omega, \end{aligned} \tag{4.1}$$

where

$$Q = HM, \tag{4.2}$$

$$H = \begin{pmatrix} rI_n & 0 & 0 \\ 0 & \beta B^T B & -B^T \\ 0 & -B & \frac{1}{\beta} I_m \end{pmatrix} \tag{4.3}$$

and

$$M = \begin{pmatrix} I_n - \frac{\beta}{r} A^T A & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & 0 & I_m \end{pmatrix}. \tag{4.4}$$

**Request on the parameter $r$** For given $\beta > 0$, $r$ should satisfy

$$\beta \|A^T A(x^k - \tilde{x}^k)\| \leq \nu r \|x^k - \tilde{x}^k\|, \quad \nu \in (0,1). \tag{4.5}$$

If the condition (2.3) is satisfied, *i.e.*, $rI_n - \beta A^T A \succ 0$, then the condition (4.5) is hold. In conversely it is not true. A conservative estimate for $\|A^T A\|$ will leads slow convergence. In the iteration process, we can check if the condition (4.5) is satisfied. This section considers the contraction in $H$-norm, where $H$ (defined in (4.3)) is symmetric and positive semi-definite.

## 4.2 Convergence of ADM-based contraction method

Based on the analysis in the last subsection, we have the following lemma.

**Lemma 4.1** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (2.2) *from the given* $w^k = (x^k, y^k, \lambda^k)$*. Then, we have*

$$(\tilde{w}^k - w^*)^T Q(w^k - \tilde{w}^k) \geq 0, \ \forall w^* \in \Omega^*, \tag{4.6}$$



*where matrix* $Q$ *is defined in* (4.2)*.*

**Proof**. Setting $w = w^*$ in (4.1), we get

$$(\tilde{w}^k - w^*)^T Q(w^k - \tilde{w}^k) \geq \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k).$$

Since $F$ is monotone, it follows that

$$\begin{aligned} & \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k) \\ \geq \quad & \theta(\tilde{u}^k) - \theta(u^*) + (\tilde{w}^k - w^*)^T F(w^*) \geq 0. \end{aligned}$$

The last inequality is due to $w^{k+1} \in \Omega$ and $w^* \in \Omega^*$ (see (1.3)). The lemma is proved. $\Box$

**Lemma 4.2** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (2.2) *from the given* $w^k = (x^k, y^k, \lambda^k)$*. Then, we have*

$$(w^k - w^*)^T HM(w^k - \tilde{w}^k) \geq \varphi(w^k, \tilde{w}^k), \ \forall\, w^* \in \Omega^*, \tag{4.7}$$

*where $H$ is defined in* (4.3) *and*

$$\varphi(w^k, \tilde{w}^k) = (w^k - \tilde{w}^k)^T HM(w^k - \tilde{w}^k) \tag{4.8}$$

**Proof**. It follows from (4.2) and (4.6) that

$$(w^k - w^*)^T HM(w^k - \tilde{w}^k) \geq (w^k - \tilde{w}^k)^T HM(w^k - \tilde{w}^k).$$

Assertion (4.7) and the definition of $\varphi(w^k, \tilde{w}^k)$ directly. $\square$

Even though $H$ is positive semi-definite, we still use $\|w - \tilde{w}\|_H$ to denote that

$$\|w - \tilde{w}\|_H = \sqrt{(w - \tilde{w})^T H(w - \tilde{w})}.$$

## 4.3 The primary contraction methods

In the primary method, we take the unit step length and use

$$w^{k+1} = w^k - M(w^k - \tilde{w}^k) \tag{4.9}$$



to update the new iterate $w^{k+1}$. According to (4.4), it can be written as

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} \tilde{x}^k + \frac{\beta}{r} A^T A(x^k - \tilde{x}^k) \\ \tilde{y}^k \\ \tilde{\lambda}^k \end{pmatrix}. \tag{4.10}$$

In the primary contraction method, only the $x$-part of the corrector is different from the predictor. In the method of Section 2, we need $r \geq \beta\|A^T A\|$. By using the method in this section, we need only a $r$ to satisfy the condition (4.5). In practical computation, we try to use the average of the eigenvalues of $\beta A^T A$.

By using (4.7), we obtain

$$\begin{aligned} &\|w^k - w^*\|_H^2 - \|w^{k+1} - w^*\|_H^2 \\ &= \; 2(w^k - w^*)^T HM(w^k - \tilde{w}^k) - \|M(w^k - \tilde{w}^k)\|_H^2 \\ &\geq \; 2\varphi(w^k, \tilde{w}^k) - \|M(w^k - \tilde{w}^k)\|_H^2. \end{aligned} \tag{4.11}$$

Note that (see (4.8))

$$\begin{aligned} &2\varphi(w^k,\tilde{w}^k) - \|M(w^k-\tilde{w}^k)\|_H^2 \\ &\quad = \quad (w^k-\tilde{w}^k)^T\big(M^TH + HM - M^THM\big)(w^k-\tilde{w}^k). \end{aligned}$$

By using the structure of the matrices $H$ and $M$, we obtain

$$M^TH + HM - M^THM = H - (I-M^T)H(I-M)$$

$$= \left(\begin{array}{ccc} rI_n & 0 & 0 \\ 0 & \beta B^TB & -B^T \\ 0 & -B & \frac{1}{\beta}I_m \end{array}\right) - \left(\begin{array}{ccc} r(\frac{\beta}{r}A^TA)^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right).$$

Therefore,

$$\begin{aligned} &2\varphi(w^k,\tilde{w}^k) - \|M(w^k-\tilde{w}^k)\|_H^2 \\ &\quad = \quad \|w^k-\tilde{w}^k\|_H^2 - r\big(\tfrac{\beta^2}{r^2}\big)\|A^TA(x^k-\tilde{x}^k)\|^2. \end{aligned}$$



Under the condition (4.5), we have

$$\big(\tfrac{\beta^2}{r^2}\big)\|A^TA(x^k-\tilde{x}^k)\|^2 \le \nu^2\|x^k-\tilde{x}^k\|^2.$$

Consequently, we have

$$2\varphi(w^k,\tilde{w}^k) - \|M(w^k-\tilde{w}^k)\|_H^2 \ge (1-\nu^2)\|w^k-\tilde{w}^k\|_H^2. \qquad (4.12)$$

**Theorem 4.1** *Let* $\tilde{w}^k = (\tilde{x}^k,\tilde{y}^k,\tilde{\lambda}^k)\in\Omega$ *be generated by* (2.2) *from the given* $w^k=(x^k,y^k,\lambda^k)$ *and the new iterate* $w^{k+1}$ *is given by* (4.9). *The sequence* $\{w^k\}$ *generated by the elementary contraction method satisfies*

$$\|w^{k+1}-w^*\|_H^2 \le \|w^k-w^*\|_H^2 - (1-\nu^2)\|w^k-\tilde{w}^k\|_H^2. \qquad (4.13)$$

Theorem 4.1 is essential for the convergence of the primary contraction method. It will lead $\lim_{k\to\infty}\|w^k-\tilde{w}^k\|_H^2 = 0$ and

$$\lim_{k\to\infty} x^k = x^*, \qquad \lim_{k\to\infty} By^k = By^* \quad \text{and} \quad \lim_{k\to\infty}\lambda^k = \lambda^*.$$

## 4.4 The general contraction methods

**The general contraction method** For given $w^k$, we use

$$w(\alpha) = w^k - \alpha M(w^k - \tilde{w}^k) \tag{4.14}$$

to update the $\alpha$-dependent new iterate. For any $w^* \in \Omega^*$, we define

$$\vartheta(\alpha) := \|w^k - w^*\|_H^2 - \|w(\alpha) - w^*\|_H^2 \tag{4.15}$$

and

$$q(\alpha) = 2\alpha\varphi(w^k, \tilde{w}^k) - \alpha^2\|M(w^k - \tilde{w}^k)\|_H^2. \tag{4.16}$$

**Theorem 4.2** *Let $w(\alpha)$ be defined by (4.14). For any $w^* \in \Omega^*$ and $\alpha \geq 0$, we have*

$$\vartheta(\alpha) \geq q(\alpha). \tag{4.17}$$



**Proof**. It follows from (4.14) and (4.15) that

$$\begin{aligned}\vartheta(\alpha) &= \|w^k - w^*\|_H^2 - \|(w^k - w^*) - \alpha M(w^k - \tilde{w}^k)\|_H^2 \\ &= 2\alpha(w^k - w^*)^T H M(w^k - \tilde{w}^k) - \alpha^2\|M(w^k - \tilde{w}^k)\|_H^2.\end{aligned}$$

By using (4.7) and the definition of $q(\alpha)$, the theorem is proved. $\Box$

Note that $q(\alpha)$ in (4.16) is a quadratic function of $\alpha$ and it reaches its maximum at

$$\alpha^* = \frac{\varphi(w^k, \tilde{w}^k)}{\|M(w^k - \tilde{w}^k)\|_H^2}. \tag{4.18}$$

From (4.12) we know that under the condition (2.3), it holds that $\alpha_k^* \geq \frac{1}{2}$. In practical computation, we use

$$w^{k+1} = w^k - \gamma\alpha_k^* M(w^k - \tilde{w}^k), \tag{4.19}$$

to update the new iterate with $\gamma \in [1, 2)$. According to (4.15) and (4.17), we have

$$\|w^{k+1} - w^*\|_H^2 \leq \|w^k - w^*\|_H^2 - q(\gamma\alpha_k^*). \tag{4.20}$$

Using (4.16) and (4.18), we obtain

$$\begin{aligned} q(\gamma\alpha_k^*) &= 2\gamma\alpha_k^*\varphi(w^k,\tilde{w}^k) - (\gamma\alpha_k^*)^2\|M(w^k-\tilde{w}^k)\|_H^2 \\ &= \gamma(2-\gamma)\alpha_k^*\varphi(w^k,\tilde{w}^k). \end{aligned} \tag{4.21}$$

Note that $\alpha_k^* > 1/2$ and (see (4.12))

$$\varphi(w^k,\tilde{w}^k) \ge \frac{1}{2}\big(\|M(w^k-\tilde{w}^k)\|_H^2 + (1-\nu)\|w^k-\tilde{w}^k\|_H^2\big). \tag{4.22}$$

Combining (4.20), (4.21) and (4.22), we get the following theorem for the general contraction method.

**Theorem 4.3** *The sequence* $\{w^k = (x^k, y^k, \lambda^k)\}$ *generated by the general contraction method* (4.19) *satisfies*

$$\begin{aligned} \|w^{k+1}-w^*\|_H^2 &\le \|w^k-w^*\|_H^2 - \frac{\gamma(2-\gamma)(1-\nu)}{4}\|w^k-\tilde{w}^k\|_H^2 \\ &\quad -\frac{\gamma(2-\gamma)}{4}\|M(w^k-\tilde{w}^k)\|_H^2. \end{aligned} \tag{4.23}$$

The inequality (4.23) in Theorem 4.3 is essential for the convergence of the



general contraction method.

On the other hand, by using (4.18) and (4.19), we have

$$\begin{aligned} q(\gamma\alpha_k^*) &= \gamma(2-\gamma)\alpha_k^*\varphi(w^k,\tilde{w}^k) \\ &= \gamma(2-\gamma)(\alpha_k^*)^2\|M(w^k-\tilde{w}^k)\|_H^2 \\ &= \frac{2-\gamma}{\gamma}\|w^k-w^{k+1}\|_H^2. \end{aligned} \tag{4.24}$$

According to (4.15), (4.17) and the above inequality, we have

**Theorem 4.4** *The sequence* $\{w^k = (x^k, y^k, \lambda^k)\}$ *generated by the general contraction method* (4.19) *satisfies*

$$\|w^{k+1}-w^*\|_H^2 \le \|w^k-w^*\|_H^2 - \frac{2-\gamma}{\gamma}\|w^k-w^{k+1}\|_H^2. \tag{4.25}$$

*Especially, by taking* $\gamma = 1$ *in* (4.19)*, then we have*

$$\|w^{k+1}-w^*\|_H^2 \le \|w^k-w^*\|_H^2 - \|w^k-w^{k+1}\|_H^2.$$

**Remark** If we dynamically take $\gamma_k = 1/\alpha_k^*$, because $\alpha^* > 1/2$, we have $\gamma_k \in (0, 2)$ and $\gamma_k \alpha_k^* \equiv 1$. In this way, we get the primary contraction method. According to (4.25), since $\gamma = 1/\alpha_k^*$, we have

$$\|w^{k+1} - w^*\|_H^2 \le \|w^k - w^*\|_H^2 - (2\alpha_k^* - 1)\|w^k - w^{k+1}\|_H^2. \tag{4.26}$$

Based on the above inequality, by using (4.18) and (4.9), we derive

$$\begin{aligned} & \|w^k - w^*\|_H^2 - \|w^{k+1} - w^*\|_H^2 \\ \ge \quad & (2\alpha_k^* - 1)\|w^k - w^{k+1}\|_H^2 \\ = \quad & \frac{2\varphi(w^k, \tilde{w}^k) - \|M(w^k - \tilde{w}^k)\|_H^2}{\|M(w^k - \tilde{w}^k)\|_H^2} \|w^k - w^{k+1}\|_H^2 \\ = \quad & 2\varphi(w^k, \tilde{w}^k) - \|M(w^k - \tilde{w}^k)\|_H^2, \end{aligned}$$

the same result as (4.11). Finally, from the last inequality, we can obtain the assertion in Theorem 4.1 for the primary contraction method.



# References


[1] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Nov. 2010.

[2] X.J. Cai, G.Y. Gu, B.S. He and X.M. Yuan, A relaxed customized proximal point algorithm for separable convex programming. http://www.optimization-online.org/DB_HTML/2011/08/3141.html

[3] D. Gabay, "Applications of the Method of Multipliers to Variational Inequalities," in *Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems*, M. Fortin and R. Glowinski, eds., North Holland, Amsterdam, The Netherlands, 299-331, 1983.

[4] D. Gabay and B. Mercier, A dual algorithm for the solution of nonlinear variational problems viafinite element approximations, Computers and Mathematics with Applications, 2:17-40, 1976.

[5] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[6] B.S. He, A class of projection and contraction methods for monotone variational inequalities, Applied Mathematics and Optimization, **35**, pp. 69-76, 1997.

[7] B.S. He, "Inexact implicit methods for monotone general variational inequalities", Math. Program. Vol. 86, Series A, pp. 199-217, 1999

[8] B.S. He, L-Z Liao, D.R. Han and H. Yang, *A new inexact alternating directions method for monotone variational inequalities*, Math. Progr., **92**, pp. 103-118, 2002

[9] B. S. He and M.-H. Xu, A general framework of contraction methods for monotone variational inequalities, Pacific J. Optimization, 4, 195-212, 2008.

[10] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Control Optim., 14 (1976), pp. 877-898.

# 凸优化和单调变分不等式的收缩算法

## 第十五讲: 三个可分离算子凸优化
## 的平行分裂增广 Lagrange 乘子法

Parallel splitting augmented Lagrangian methods for
convex optimization with three separable operators

南京大学数学系　　何炳生
hebma@nju.edu.cn

The context of this lecture is based on the publication [6]



## 1 Convex Optimization with 3 separable operators

We consider the linearly constrained convex optimization with 3 separable operators:

$$\min\{\theta_1(x)+\theta_2(y)+\theta_3(z)\,|\,Ax+By+Cz=b,\ x\in\mathcal{X},\ y\in\mathcal{Y},\ z\in\mathcal{Z}\} \quad (1.1)$$

where $\theta_1(x):\Re^{n_1}\to\Re,\ \theta_2(y):\Re^{n_2}\to\Re,\ \theta_3(z):\Re^{n_3}\to\Re$ are convex; $A\in\Re^{m\times n_1}$, $B\in\Re^{m\times n_2}$, $C\in\Re^{m\times n_3}$, $b\in\Re^m$; $\mathcal{X}\subset\Re^{n_1}$, $\mathcal{Y}\subset\Re^{n_2}$ and $\mathcal{Z}\subset\Re^{n_3}$ are given convex set. Let $n=n_1+n_2+n_3$.

This optimization problem is equivalent to find $(x^*,y^*,z^*,\lambda^*)\in\Omega$, such that

$$\left\{\begin{array}{l}\theta_1(x)-\theta_1(x^*)+(x-x^*)^T(-A^T\lambda^*)\ge 0,\\ \theta_2(y)-\theta_2(y^*)+(y-y^*)^T(-B^T\lambda^*)\ge 0,\\ \theta_3(z)-\theta_3(z^*)+(z-z^*)^T(-C^T\lambda^*)\ge 0,\\ (\lambda-\lambda^*)^T(Ax^*+By^*+Cz^*-b)\ge 0,\end{array}\right. \quad \forall\,(x,y,z,\lambda)\in\Omega, \qquad (1.2)$$

where

$$\Omega=\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}\times\Re^m.$$

By denoting

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

and

$$\theta(u)=\theta_1(x)+\theta_2(y)+\theta_3(z),$$

the optimal condition (1.2) can be written as

$$\boxed{w^*\in\Omega,\quad \theta(u)-\theta(u^*)+(w-w^*)^TF(w^*)\ge 0,\quad \forall\, w\in\Omega.}$$

For convenience we use the notations

$$v=\begin{pmatrix} y\\ z\\ \lambda \end{pmatrix},\qquad \text{and}\qquad \mathcal{V}^*=\{(y^*,z^*,\lambda^*)\,|\,(x^*,y^*,z^*,\lambda^*)\in\Omega^*\}.$$



## 2 Parallel Splitting Augmented Lagrangian Method

PSALM is a prediction-correction method. First it generates a predictor:

1. From given $(x^k,y^k,z^k,\lambda^k)$, obtain $\tilde{x}^k$, $\tilde{y}^k$ and $\tilde{z}^k$, by the following parallel manner:

$$\tilde{x}^k=\text{Argmin}\left\{\begin{array}{c}\theta_1(x)-(\lambda^k)^T(Ax+By^k+Cz^k-b)\\ +\frac{\beta}{2}\|Ax+By^k+Cz^k-b\|^2\end{array}\middle|\, x\in\mathcal{X}\right\}\tag{2.1a}$$

$$\tilde{y}^k=\text{Argmin}\left\{\begin{array}{c}\theta_2(y)-(\lambda^k)^T(Ax^k+By+Cz^k-b)\\ +\frac{\beta}{2}\|Ax^k+By+Cz^k-b\|^2\end{array}\middle|\, y\in\mathcal{Y}\right\}\tag{2.1b}$$

$$\tilde{z}^k=\text{Argmin}\left\{\begin{array}{c}\theta_3(z)-(\lambda^k)^T(Ax^k+By^k+Cz-b)\\ +\frac{\beta}{2}\|Ax^k+By^k+Cz-b\|^2\end{array}\middle|\, z\in\mathcal{Z}\right\}\tag{2.1c}$$

2. Update $\tilde{\lambda}^k$ by

$$\tilde{\lambda}^k=\lambda^k-\beta(A\tilde{x}^k+B\tilde{y}^k+C\tilde{z}^k-b).\tag{2.1d}$$

Note that the solution $(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ of (2.1a)-(2.1c) satisfies

$$\begin{cases} \theta_1(x) - \theta_1(\tilde{x}^k) + (x - \tilde{x}^k)^T \{-A^T \lambda^k + \beta A^T (A\tilde{x}^k + By^k + Cz^k - b)\} \ge 0 \\ \theta_2(y) - \theta_2(\tilde{y}^k) + (y - \tilde{y}^k)^T \{-B^T \lambda^k + \beta B^T (Ax^k + B\tilde{y}^k + Cz^k - b)\} \ge 0 \\ \theta_3(z) - \theta_3(\tilde{z}^k) + (z - \tilde{z}^k)^T \{-C^T \lambda^k + \beta C^T (Ax^k + By^k + C\tilde{z}^k - b)\} \ge 0 \end{cases} \tag{2.2}$$

for all $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. Using

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b)$$

and by a manipulation, (2.2) can be rewritten as $(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$,

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \\ z - \tilde{z}^k \end{pmatrix}^T \begin{pmatrix} -A^T \tilde{\lambda}^k + \beta A^T [B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)] \\ -B^T \tilde{\lambda}^k + \beta B^T [A(x^k - \tilde{x}^k) + C(z^k - \tilde{z}^k)] \\ -C^T \tilde{\lambda}^k + \beta C^T [A(x^k - \tilde{x}^k) + B(y^k - \tilde{y}^k)] \end{pmatrix} \ge 0, \tag{2.3}$$

for all $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. Combining the last inequality with (2.1d), we



have

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \\ z - \tilde{z}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T \tilde{\lambda}^k \\ -B^T \tilde{\lambda}^k \\ -C^T \tilde{\lambda}^k \\ A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b \end{pmatrix} \right.$$

$$+ \beta \begin{pmatrix} A^T \\ B^T \\ C^T \\ 0 \end{pmatrix} \big( A(x^k - \tilde{x}^k) + B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k) \big)$$

$$\left. + \begin{pmatrix} \beta A^T A & 0 & 0 & 0 \\ 0 & \beta B^T B & 0 & 0 \\ 0 & 0 & \beta C^T C & 0 \\ 0 & 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} \tilde{x}^k - x^k \\ \tilde{y}^k - y^k \\ \tilde{z}^k - z^k \\ \tilde{\lambda}^k - \lambda^k \end{pmatrix} \right\} \ge 0, \ \forall w \in \Omega. \tag{2.4}$$

Based on the above analysis, we have the following lemma.

**Lemma 2.1** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (2.1) *from the given* $w^k = (x^k, y^k, z^k, \lambda^k)$. *Then, we have*

$$(\tilde{w}^k - w^*)^T H(w^k - \tilde{w}^k) \geq (\tilde{w}^k - w^*)^T \big(F(\tilde{w}^k) + \eta(u^k, \tilde{u}^k)\big), \tag{2.5}$$

*where*

$$\eta(u^k, \tilde{u}^k) = \beta \begin{pmatrix} A^T \\ B^T \\ C^T \\ 0 \end{pmatrix} \big[A(x^k - \tilde{x}^k) + B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big] \tag{2.6}$$

*and*

$$H = \begin{pmatrix} \beta A^T A & 0 & 0 & 0 \\ 0 & \beta B^T B & 0 & 0 \\ 0 & 0 & \beta C^T C & 0 \\ 0 & 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix}. \tag{2.7}$$



**Proof**. Setting $(x, y, z, \lambda) = (x^*, y^*, z^*, \lambda^*)$ in (2.4), the assertion follows directly. $\Box$

Since $F$ is monotone and $\tilde{w}^k \in \Omega$, it follows that

$$(\tilde{w}^k - w^*)^T F(\tilde{w}^k) \geq (\tilde{w}^k - w^*)^T F(w^*) \geq 0. \tag{2.8}$$

In addition, by using $Ax^* + By^* + Cz^* = b$ and

$$\beta(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b) = (\lambda^k - \tilde{\lambda}^k),$$

we have

$$\begin{aligned} &(\tilde{w}^k - w^*)^T \eta(u^k, \tilde{u}^k) \\ &\quad = (A(x^k - \tilde{x}^k) + B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k))^T \beta(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b) \\ &\quad = (\lambda^k - \tilde{\lambda}^k)^T [A(x^k - \tilde{x}^k) + B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)]. \end{aligned} \tag{2.9}$$

**Lemma 2.2** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (2.1) *from the* *given* $w^k = (x^k, y^k, z^k, \lambda^k)$. *Then, we have*

$$(w^k - w^*)^T H(w^k - \tilde{w}^k) \geq \varphi(w^k, \tilde{w}^k), \quad \forall\, w^* \in \Omega^*, \tag{2.10}$$

*where*

$$\begin{aligned}\varphi(w^k, \tilde{w}^k) &= \|w^k - \tilde{w}^k\|_H^2 \\ &\quad + (\lambda^k - \tilde{\lambda}^k)^T \big(A(x^k - \tilde{x}^k) + B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big).\end{aligned} \tag{2.11}$$

**Proof**. First, using (2.5), (2.8) and (2.9) we obtain that

$$\begin{aligned}&(\tilde{w}^k - w^*)^T H(w^k - \tilde{w}^k) \\ &\quad \geq \quad (\lambda^k - \tilde{\lambda}^k)^T \big(A(x^k - \tilde{x}^k) + B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big).\end{aligned}$$

The assertion of this lemma follows from the last inequality and the definition of $\varphi(w^k, \tilde{w}^k)$ (2.11) directly. $\square$

Now, we consider the right hand side of (2.10). Note that



$$\begin{aligned}\varphi(w^k, \tilde{w}^k) &= \|w^k - \tilde{w}^k\|_H^2 \\ &\quad + (\lambda^k - \tilde{\lambda}^k)^T \big(A(x^k - \tilde{x}^k) + B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big) \\ &= \begin{pmatrix} x^k - \tilde{x}^k \\ y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} \beta A^T A & 0 & 0 & \frac{1}{2}A^T \\ 0 & \beta B^T B & 0 & \frac{1}{2}B^T \\ 0 & 0 & \beta C^T C & \frac{1}{2}C^T \\ \frac{1}{2}A & \frac{1}{2}B & \frac{1}{2}C & \frac{1}{\beta}I_m \end{pmatrix} \begin{pmatrix} x^k - \tilde{x}^k \\ y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \\ &= \begin{pmatrix} \sqrt{\beta}A(x^k - \tilde{x}^k) \\ \sqrt{\beta}B(y^k - \tilde{y}^k) \\ \sqrt{\beta}C(z^k - \tilde{z}^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}^T \begin{pmatrix} I_m & 0 & 0 & \frac{1}{2}I_m \\ 0 & I_m & 0 & \frac{1}{2}I_m \\ 0 & 0 & I_m & \frac{1}{2}I_m \\ \frac{1}{2}I_m & \frac{1}{2}I_m & \frac{1}{2}I_m & I_m \end{pmatrix} \begin{pmatrix} \sqrt{\beta}A(x^k - \tilde{x}^k) \\ \sqrt{\beta}B(y^k - \tilde{y}^k) \\ \sqrt{\beta}C(z^k - \tilde{z}^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}.\end{aligned} \tag{2.12}$$

Because the eigenvalues of the matrix

$$\begin{pmatrix} 1 & 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix} \quad \text{are } 1,\ 1 \text{ and } 1 \pm \frac{\sqrt{3}}{2},$$

The smallest eigenvalue is $\frac{2-\sqrt{3}}{2}$. Therefore,

$$\begin{aligned} \varphi(w^k, \tilde{w}^k) \quad &\geq \quad \frac{2-\sqrt{3}}{2} \begin{pmatrix} \sqrt{\beta}A(x^k - \tilde{x}^k) \\ \sqrt{\beta}B(y^k - \tilde{y}^k) \\ \sqrt{\beta}C(z^k - \tilde{z}^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}^T \begin{pmatrix} \sqrt{\beta}A(x^k - \tilde{x}^k) \\ \sqrt{\beta}B(y^k - \tilde{y}^k) \\ \sqrt{\beta}C(z^k - \tilde{z}^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix} \\ &= \quad \frac{2-\sqrt{3}}{2} \|w^k - \tilde{w}^k\|_H^2. \end{aligned} \tag{2.13}$$



**Correction:** Based on the predictor by (2.1), update the new iterate $w^{k+1}$ by:

$$w^{k+1} = w^k - \alpha_k(w^k - \tilde{w}^k), \qquad \alpha_k = \gamma\alpha_k^*, \qquad \gamma \in (0,2) \tag{2.14a}$$

where

$$\alpha_k^* = \frac{\varphi(w^k, \tilde{w}^k)}{\|w^k - \tilde{w}^k\|_H^2}. \tag{2.14b}$$

By using (2.10) and (2.14), we obtain

$$\begin{aligned} \|w^{k+1} - w^*\|_H^2 \quad &= \quad \|(w^k - w^*) - \gamma\alpha_k^*(w^k - \tilde{w}^k)\|_H^2 \\ &= \quad \|w^k - w^*\|_H^2 - 2\gamma\alpha_k^*(w^k - w^*)^T H(w^k - \tilde{w}^k) \\ &\qquad +\gamma^2(\alpha_k^*)^2\|w^k - \tilde{w}^k\|_H^2 \\ &\leq \quad \|w^k - w^*\|_H^2 - 2\gamma\alpha_k^*\varphi(w^k, \tilde{w}^k) \\ &\qquad +\gamma^2(\alpha_k^*)^2\|w^k - \tilde{w}^k\|_H^2 \\ &\leq \quad \|w^k - w^*\|_H^2 - \gamma(2-\gamma)\alpha_k^*\varphi(w^k, \tilde{w}^k). \end{aligned} \tag{2.15}$$

In comparison with the computational load for obtaining $(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k)$, the calculation cost for step-size $\alpha_k^*$ is slight.

**Convergence** Using (2.13) and (2.14b), we obtain

$$\|w^{k+1} - w^*\|_H^2 \le \|w^k - w^*\|_H^2 - \frac{\gamma(2-\gamma)(7-4\sqrt{3})}{4}\|w^k - \tilde{w}^k\|_H^2.$$

♣ B. S. He, Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities, Computational Optimization and Applications, 42, 195-212, 2009.

# 3 Partially Parallel Splitting ALM

In the partially parallel splitting ALM, $x$ is an intermediate variable. The iteration begins with $v^k = (y^k, z^k, \lambda^k)$ and produces new iterate $v^{k+1}$. It is still a prediction-correction method.



**Prediction:**

1. From given $(y^k, z^k, \lambda^k)$, obtain $\tilde{x}^k, \tilde{y}^k$ and $\tilde{z}^k$, by the following partially parallel manner:

$$\tilde{x}^k = \text{Argmin}\left\{ \begin{array}{l} \theta_1(x) - (\lambda^k)^T(Ax + By^k + Cz^k - b) \\ +\frac{\beta}{2}\|Ax + By^k + Cz^k - b\|^2 \end{array} \middle| x \in \mathcal{X} \right\} \quad (3.1a)$$

$$\tilde{y}^k = \text{Argmin}\left\{ \begin{array}{l} \theta_2(y) - (\lambda^k)^T(A\tilde{x}^k + By + Cz^k - b) \\ +\frac{\beta}{2}\|A\tilde{x}^k + By + Cz^k - b\|^2 \end{array} \middle| y \in \mathcal{Y} \right\} \quad (3.1b)$$

$$\tilde{z}^k = \text{Argmin}\left\{ \begin{array}{l} \theta_3(z) - (\lambda^k)^T(A\tilde{x}^k + By^k + Cz - b) \\ +\frac{\beta}{2}\|A\tilde{x}^k + By^k + Cz - b\|^2 \end{array} \middle| z \in \mathcal{Z} \right\} \quad (3.1c)$$

2. Update $\tilde{\lambda}^k$ by

$$\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b). \quad (3.1d)$$

Note that in (3.1b) and (3.1c), we use $\tilde{x}^k$ generated by (3.1a).

**Analysis** The solution $(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ of (3.1a)-(3.1c) satisfies

$$\begin{cases} \theta_1(x) - \theta_1(\tilde{x}^k) + (x - \tilde{x}^k)^T \{-A^T\lambda^k + \beta A^T\big(A\tilde{x}^k + By^k + Cz^k - b\big)\} \geq 0 \\ \theta_2(y) - \theta_2(\tilde{y}^k) + (y - \tilde{y}^k)^T \{-B^T\lambda^k + \beta B^T\big(A\tilde{x}^k + B\tilde{y}^k + Cz^k - b\big)\} \geq 0 \\ \theta_3(z) - \theta_3(\tilde{z}^k) + (z - \tilde{z}^k)^T \{-C^T\lambda^k + \beta C^T\big(A\tilde{x}^k + By^k + C\tilde{z}^k - b\big)\} \geq 0 \end{cases} \tag{3.2}$$

for all $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. Using $\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b)$ and by a manipulation, (3.2) can be rewritten as $(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$,

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \\ z - \tilde{z}^k \end{pmatrix}^T \begin{pmatrix} -A^T\tilde{\lambda}^k + \beta A^T[B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)] \\ -B^T\tilde{\lambda}^k + \beta B^T[\quad 0 \quad + C(z^k - \tilde{z}^k)] \\ -C^T\tilde{\lambda}^k + \beta C^T[B(y^k - \tilde{y}^k) + \quad 0 \quad ] \end{pmatrix} \geq 0,$$

for all $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. Combining the last inequality with (3.1d), we



have

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \\ z - \tilde{z}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ -B^T\tilde{\lambda}^k \\ -C^T\tilde{\lambda}^k \\ A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b \end{pmatrix} \right.$$

$$+\beta \begin{pmatrix} A^T \\ B^T \\ C^T \\ 0 \end{pmatrix} \big(B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big)$$

$$\left. + \begin{pmatrix} 0 & 0 & 0 \\ \beta B^T B & 0 & 0 \\ 0 & \beta C^T C & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} \tilde{y}^k - y^k \\ \tilde{z}^k - z^k \\ \tilde{\lambda}^k - \lambda \end{pmatrix} \right\} \geq 0, \ \forall w \in \Omega. \tag{3.3}$$

Based on the above analysis, we have the following lemma.

**Lemma 3.1** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (3.1) *from the given* $v^k = (y^k, z^k, \lambda^k)$*. Then, we have*

$$(\tilde{v}^k - v^*)^T H(v^k - \tilde{v}^k) \geq (\tilde{w}^k - w^*)^T \big(F(\tilde{w}^k) + \eta(u^k, \tilde{u}^k)\big), \tag{3.4}$$

*where*

$$\eta(u^k, \tilde{u}^k) = \beta \begin{pmatrix} A^T \\ B^T \\ C^T \\ 0 \end{pmatrix} \big(B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big) \tag{3.5}$$

*and*

$$H = \begin{pmatrix} \beta B^T B & 0 & 0 \\ 0 & \beta C^T C & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix}. \tag{3.6}$$

**Proof**. Setting $(x, y, z, \lambda) = (x^*, y^*, z^*, \lambda^*)$ in (3.3), and using $v = (y, z, \lambda)$ the assertion follows directly. $\Box$



Since $F$ is monotone and $\tilde{w}^k \in \Omega$, it follows that

$$(\tilde{w}^k - w^*)^T F(\tilde{w}^k) \geq (\tilde{w}^k - w^*)^T F(w^*) \geq 0. \tag{3.7}$$

In addition, by using $Ax^* + By^* + Cz^* = b$ and

$$\beta(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b) = (\lambda^k - \tilde{\lambda}^k),$$

we have

$$\begin{aligned} & \eta(u^k, \tilde{u}^k)^T (\tilde{w}^k - w^*) \\ & \quad = \quad (B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k))^T \beta(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b) \\ & \quad = \quad (B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k))^T (\lambda^k - \tilde{\lambda}^k). \end{aligned} \tag{3.8}$$

**Lemma 3.2** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (3.1) *from the given* $v^k = (y^k, z^k, \lambda^k)$*. Then, we have*

$$(v^k - v^*)^T H(v^k - \tilde{v}^k) \geq \varphi(v^k, \tilde{v}^k), \quad \forall\, w^* \in \Omega^*, \tag{3.9}$$

*where*

$$\varphi(v^k, \tilde{v}^k) = \|v^k - \tilde{v}^k\|_H^2 + (\lambda^k - \tilde{\lambda}^k)^T \big(B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big). \tag{3.10}$$

**Proof**. First, using (3.4), (3.7) and (3.8) we obtain that

$$(\tilde{v}^k - v^*)^T H(v^k - \tilde{v}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T \big(B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big).$$

The assertion of this lemma follows from the last inequality and the definition of $\varphi(v^k, \tilde{v}^k)$ directly. $\Box$

**Lemma 3.3** *Let* $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (3.1) *from the given* $v^k = (y^k, z^k, \lambda^k)$*. For* $H$ *defined in* (3.6) *and* $\varphi(v^k, \tilde{v}^k)$ *defined in* (3.10)*, we have*

$$\varphi(v^k, \tilde{v}^k) \geq \frac{2 - \sqrt{2}}{2} \|v^k - \tilde{v}^k\|_H^2. \tag{3.11}$$



**Proof**. According to the definition of (3.10), we obtain

$$\varphi(v^k, \tilde{v}^k) = \|v^k - \tilde{v}^k\|_H^2 + (\lambda^k - \tilde{\lambda}^k)^T \big(B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big)$$

$$= \begin{pmatrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} \beta B^T B & 0 & \frac{1}{2} B^T \\ 0 & \beta C^T C & \frac{1}{2} C^T \\ \frac{1}{2} B & \frac{1}{2} C & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}$$

$$= \begin{pmatrix} \sqrt{\beta} B(y^k - \tilde{y}^k) \\ \sqrt{\beta} C(z^k - \tilde{z}^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}^T \begin{pmatrix} I_m & 0 & \frac{1}{2} I_m \\ 0 & I_m & \frac{1}{2} I_m \\ \frac{1}{2} I_m & \frac{1}{2} I_m & I_m \end{pmatrix} \begin{pmatrix} \sqrt{\beta} B(y^k - \tilde{y}^k) \\ \sqrt{\beta} C(z^k - \tilde{z}^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}.$$

Because the matrix

$$\begin{pmatrix} 1 & 0 & \frac{1}{2} \\ 0 & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix}$$

is positive definite and its eigenvalues are $1$ and $1 \pm \frac{\sqrt{2}}{2}$, thus, we obtain

$$\begin{aligned}\varphi(v^k, \tilde{v}^k) &\geq \frac{2-\sqrt{2}}{2} \begin{pmatrix} \sqrt{\beta}B(y^k - \tilde{y}^k) \\ \sqrt{\beta}C(z^k - \tilde{z}^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}^T \begin{pmatrix} \sqrt{\beta}B(y^k - \tilde{y}^k) \\ \sqrt{\beta}C(z^k - \tilde{z}^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix} \\ &= \frac{2-\sqrt{2}}{2}\|v^k - \tilde{v}^k\|_H^2,\end{aligned}$$

and the assertion is proved. $\Box$

**Correction** Based on the predictor by (3.1), update the new iterate $v^{k+1}$ by

$$v^{k+1} = v^k - \alpha_k(v^k - \tilde{v}^k), \qquad \alpha_k = \gamma\alpha_k^*, \qquad \gamma \in (0,2) \tag{3.12a}$$

where

$$\alpha_k^* = \frac{\varphi(v^k, \tilde{v}^k)}{\|v^k - \tilde{v}^k\|_H^2} \tag{3.12b}$$



By using (3.9) and (3.12), we obtain

$$\begin{aligned}\|v^{k+1} - v^*\|_H^2 &= \|(v^k - v^*) - \gamma\alpha_k^*(v^k - \tilde{v}^k)\|_H^2 \\ &= \|v^k - v^*\|_H^2 - 2\gamma\alpha_k^*(v^k - v^*)^T H(v^k - \tilde{v}^k) + \gamma^2(\alpha_k^*)^2\|v^k - \tilde{v}^k\|_H^2 \\ &\leq \|v^k - v^*\|_H^2 - 2\gamma\alpha_k^*\varphi(v^k, \tilde{v}^k) + \gamma^2(\alpha_k^*)^2\|v^k - \tilde{v}^k\|_H^2 \\ &= \|v^k - v^*\|_H^2 - \gamma(2-\gamma)\alpha_k^*\varphi(v^k, \tilde{v}^k). \end{aligned} \tag{3.13}$$

In comparison with the computational load for obtaining $(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k)$, the calculation cost for step-size $\alpha_k^*$ is slight.

**Convergence** Using (3.11) and (3.12b), it follows from (3.13) that

$$\|v^{k+1} - v^*\|_H^2 \leq \|v^k - v^*\|_H^2 - \frac{\gamma(2-\gamma)(3-2\sqrt{2})}{2}\|v^k - \tilde{v}^k\|_H^2.$$

From (3.11) we know that $\alpha_k^* \geq \frac{2-\sqrt{2}}{2}$. In the correction step, we can also take the fixed step size $\alpha_k \equiv \alpha \in (0, 2-\sqrt{2})$, the method is still convergent.

# 4 Extension to problems with 4 operators

We consider the following optimization problem with 4 separable operators:

$$\min\{\sum_{i=1}^{4}\theta_i(x_i) \mid \sum_{i=1}^{4} A_i x_i = b, x_i \in \mathcal{X}_i\} \tag{4.1}$$

where $\theta_i(x_i): \Re^{n_i} \to \Re$, is convex, $A_i \in \Re^{m\times n_i}$, $\mathcal{X}_i \subset \Re^{n_i}$ is given convex set, $i = 1,\ldots,4$; $b \in \Re^m$. This optimization problem is equivalent to find $w^* = (x_1^*, x_2^*, x_3^*, x_4^*, \lambda^*) \in \Omega$, such that

$$\left\{\begin{array}{l} \theta_1(x_1) - \theta_1(x_1^*) + (x_1 - x_1^*)^T(-A_1^T\lambda^*) \geq 0, \\ \theta_2(x_2) - \theta_2(x_2^*) + (x_2 - x_2^*)^T(-A_2^T\lambda^*) \geq 0, \\ \theta_3(x_3) - \theta_3(x_3^*) + (x_3 - x_3^*)^T(-A_4^T\lambda^*) \geq 0, \\ \theta_4(x_4) - \theta_4(x_4^*) + (x_4 - x_4^*)^T(-A_4^T\lambda^*) \geq 0, \\ (\lambda - \lambda^*)^T(A_1x_1^* + A_2x_2^* + A_3x_3^* + A_4x_4^* - b) \geq 0, \end{array}\right. \quad \forall\, w \in \Omega, \tag{4.2}$$

where

$$\Omega = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \times \mathcal{X}_4 \times \Re^m.$$



By denoting

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}, \quad w = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \lambda \end{pmatrix} \quad F(w) = \begin{pmatrix} -A_1^T\lambda \\ -A_2^T\lambda \\ -A_3^T\lambda \\ -A_4^T\lambda \\ \sum_{i=1}^4 A_i x_i - b, \end{pmatrix},$$

and $\quad \theta(x) = \theta_1(x_1) + \theta_2(x_2) + \theta_3(x_3) + \theta_4(x_4),$

the optimal condition (4.2) can be written as

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

**Full parallel splitting augmented Lagrangian method**

If we extend the full parallel splitting augmented Lagrangian method to solve (4.1), similarly as in Lemma 2.2 (see (2.10) and (2.11)), we have

$$(w^k - w^*)^T H(w^k - \tilde{w}^k) \geq \varphi(w^k, \tilde{w}^k), \quad \forall\, w^* \in \Omega^*, \tag{4.3}$$

where

$$\varphi(w^k, \tilde{w}^k) = \|w^k - \tilde{w}^k\|_H^2 + (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{i=1}^{4} A_i(x_i^k - \tilde{x}_i^k)\big) \qquad (4.4)$$

and

$$H = \textbf{diag}(\beta A_1^T A_1, \beta A_2^T A_2, \beta A_3^T A_3, \beta A_4^T A_4, \frac{1}{\beta} I_m).$$

Note that, similar as (2.12), we have

$$\varphi(w^k, \tilde{w}^k) = \begin{pmatrix} \sqrt{\beta}A_1(x_1^k - \tilde{x}_1^k) \\ \sqrt{\beta}A_2(x_2^k - \tilde{x}_2^k) \\ \sqrt{\beta}A_3(x_3^k - \tilde{x}_3^k) \\ \sqrt{\beta}A_4(x_4^k - \tilde{x}_4^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}^T \begin{pmatrix} I & 0 & 0 & 0 & I/2 \\ 0 & I & 0 & 0 & I/2 \\ 0 & 0 & I & 0 & I/2 \\ 0 & 0 & 0 & I & I/2 \\ I/2 & I/2 & I/2 & I/2 & I \end{pmatrix} \begin{pmatrix} \sqrt{\beta}A_1(x_1^k - \tilde{x}_1^k) \\ \sqrt{\beta}A_2(x_2^k - \tilde{x}_2^k) \\ \sqrt{\beta}A_3(x_3^k - \tilde{x}_3^k) \\ \sqrt{\beta}A_4(x_4^k - \tilde{x}_4^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}.$$

In other words, for the problem with 4 separable operators, if we use the full



parallel splitting augmented Lagrangian method, we will met a matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix}.$$

The above matrix is positive semi-definite and its eigenvalues are $0, 1, 1, 1$ and $2$. The vector $(1, 1, 1, 1, -2)^T$ is the eigenvector related to the eigenvalue 0. This means, even through $\|w^k - \tilde{w}^k\|_H \neq 0$, it can not guarantee that $\varphi(w^k, \tilde{w}^k) > 0$. Therefore, we suggest to use the partially splitting augmented Lagrangian method.

**Partially parallel splitting augmented Lagrangian method**

In the partially parallel splitting augmented Lagrangian method, $x_1$ is an intermediate variable during the iterations. After solving the $x_1$-subproblem, then solve the $x_2$, $x_3$, $x_4$-subproblems in the parallel manner !

**Prediction** Similarly as the prediction step (3.1) for problems with 3 operators

From given $(x_2^k, x_3^k, x_4^k, \lambda^k)$, obtain $\tilde{w}^k$ by the following partially parallel manner:

$$\tilde{x}_1^k = \text{Argmin}\left\{ \begin{array}{l} \theta_1(x_1) - (\lambda^k)^T(A_1x_1 + A_2x_2^k + A_3x_3^k + A_4x_4^k - b) \\ \quad + \frac{\beta}{2}\|A_1x_1 + A_2x_2^k + A_3x_3^k + A_4x_4^k - b\|^2 \end{array} \middle| x_1 \in \mathcal{X}_1 \right\} \tag{4.5a}$$

$$\tilde{x}_2^k = \text{Argmin}\left\{ \begin{array}{l} \theta_2(x_2) - (\lambda^k)^T(A_1\tilde{x}_1^k + A_2x_2 + A_3x_3^k + A_4x_4^k - b) \\ \quad + \frac{\beta}{2}\|A_1\tilde{x}_1^k + A_2x_2 + A_3x_3^k + A_4x_4^k - b\|^2 \end{array} \middle| x_2 \in \mathcal{X}_2 \right\} \tag{4.5b}$$

$$\tilde{x}_3^k = \text{Argmin}\left\{ \begin{array}{l} \theta_3(x_3) - (\lambda^k)^T(A_1\tilde{x}_1^k + A_2x_2^k + A_3x_3 + A_4x_4^k - b) \\ \quad + \frac{\beta}{2}\|A_1\tilde{x}_1^k + A_2x_2^k + A_3x_3 + A_4x_4^k - b\|^2 \end{array} \middle| x_3 \in \mathcal{X}_3 \right\} \tag{4.5c}$$

$$\tilde{x}_4^k = \text{Argmin}\left\{ \begin{array}{l} \theta_4(x_4) - (\lambda^k)^T(A_1\tilde{x}_1^k + A_2x_2^k + A_3x_3^k + A_4x_4 - b) \\ \quad + \frac{\beta}{2}\|A_1\tilde{x}_1^k + A_2x_2^k + A_3x_3^k + A_4x_4 - b\|^2 \end{array} \middle| x_4 \in \mathcal{X}_4 \right\} \tag{4.5d}$$

Update $\tilde{\lambda}^k$ by

$$\tilde{\lambda}^k = \lambda^k - \beta(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k + A_3\tilde{x}_3^k + A_4\tilde{x}_4^k - b). \tag{4.5e}$$

Note that in (4.5b), (4.5c) and (4.5d), we use $\tilde{x}_1^k$ generated by (4.5a).



Similar as (3.3), for the $\tilde{w}^k$ generated by (4.5), we have

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ x_3 - \tilde{x}_3^k \\ x_4 - \tilde{x}_4^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A_1^T\tilde{\lambda}^k \\ -A_2^T\tilde{\lambda}^k \\ -A_3^T\tilde{\lambda}^k \\ -A_4^T\tilde{\lambda}^k \\ A_1\tilde{x}_1^k + A_2\tilde{x}_2^k + A_3\tilde{x}_3^k + A_4\tilde{x}_4^k - b \end{pmatrix} \right.$$

$$+\beta \begin{pmatrix} A_1^T \\ A_2^T \\ A_3^T \\ A_4^T \\ 0 \end{pmatrix} \left[ A_2(x_2^k - \tilde{x}_2^k) + A_3(x_3^k - \tilde{x}_3^k) + A_4(x_4^k - \tilde{x}_4^k) \right]$$

$$\left. + \begin{pmatrix} 0 & 0 & 0 & 0 \\ \beta A_2^TA_2 & 0 & 0 & 0 \\ 0 & \beta A_3^TA_3 & 0 & 0 \\ 0 & 0 & \beta A_4^TA_4 & 0 \\ 0 & 0 & 0 & \frac{1}{\beta}I_m \end{pmatrix} \begin{pmatrix} \tilde{x}_2^k - x_2^k \\ \tilde{x}_3^k - x_3^k \\ \tilde{x}_4^k - x_4^k \\ \tilde{\lambda}^k - \lambda \end{pmatrix} \right\} \geq 0, \ \forall w \in \Omega. \tag{4.6}$$

For convenience we use the notations

$$v = \begin{pmatrix} x_2 \\ x_3 \\ x_4 \\ \lambda \end{pmatrix} \quad \text{and} \quad \mathcal{V}^* = \{(x_2^*, x_3^*, x_4^*, \lambda^*) \,|\, (x_1^*, x_2^*, x_3^*, x_4^*, \lambda^*) \in \Omega^*\}.$$

Based on the above analysis, we have the following lemma.

**Lemma 4.1** *Let* $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \tilde{x}_3^k, \tilde{x}_4^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (4.5) *from the given* $v^k = (x_2^k, x_3^k, x_4^k, \lambda^k)$. *Then, we have*

$$(\tilde{v}^k - v^*)^T H(v^k - \tilde{v}^k) \geq (\tilde{w}^k - w^*)^T \big(F(\tilde{w}^k) + \eta(x^k, \tilde{x}^k)\big), \tag{4.7}$$

*where*

$$H = \textbf{diag}(\beta A_2^T A_2, \beta A_3^T A_3, \beta A_4^T A_4, \frac{1}{\beta} I_m) \tag{4.8}$$



*and*

$$\eta(x^k, \tilde{x}^k) = \beta \begin{pmatrix} A_1^T \\ A_2^T \\ A_3^T \\ A_4^T \\ 0 \end{pmatrix} \big[A_2(x_2^k - \tilde{x}_2^k) + A_3(x_3^k - \tilde{x}_3^k) + A_4(x_4^k - \tilde{x}_4^k)\big] \tag{4.9}$$

**Lemma 4.2** *Let* $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \tilde{x}_3^k, \tilde{x}_4^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (4.5) *from the given* $v^k = (x_2^k, x_3^k, x_4^k, \lambda^k)$. *Then, we have*

$$(v^k - v^*)^T H(v^k - \tilde{v}^k) \geq \varphi(v^k, \tilde{v}^k), \quad \forall\, v^* \in \mathcal{V}^*, \tag{4.10}$$

*where*

$$\begin{aligned} \varphi(v^k, \tilde{v}^k) &= \|v^k - \tilde{v}^k\|_H^2 \\ &\quad + (\lambda^k - \tilde{\lambda}^k)^T \big[A_2(x_2^k - \tilde{x}_2^k) + A_3(x_3^k - \tilde{x}_3^k) + A_4(x_4^k - \tilde{x}_4^k)\big]. \end{aligned} \tag{4.11}$$

The proofs of Lemma 4.2 is similar as those in Lemma 3.2.

**Lemma 4.3** *Let* $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \tilde{x}_3^k, \tilde{x}_4^k, \tilde{\lambda}^k) \in \Omega$ *be generated by* (4.5) *from the given* $v^k = (x_2^k, x_3^k, x_4^k, \lambda^k)$. *For* $H$ *defined in* (4.8) *and* $\varphi(v^k, \tilde{v}^k)$ *defined in* (4.11), *we have*

$$\varphi(v^k, \tilde{v}^k) \geq \frac{2-\sqrt{3}}{2} \|v^k - \tilde{v}^k\|_H^2. \tag{4.12}$$

**Proof**. According to the definition of (4.11), we have

$$\begin{aligned}
\varphi(v^k, \tilde{v}^k) &= \|v^k - \tilde{v}^k\|_H^2 \\
&\quad + (\lambda^k - \tilde{\lambda}^k)^T \big[A_2(x_2^k - \tilde{x}_2^k) + A_3(x_3^k - \tilde{x}_3^k) + A_4(x_4^k - \tilde{x}_4^k)\big] \\
&= \begin{pmatrix} \sqrt{\beta}A_2(x_2^k - \tilde{x}_2^k) \\ \sqrt{\beta}A_3(x_3^k - \tilde{x}_3^k) \\ \sqrt{\beta}A_4(x_4^k - \tilde{x}_4^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}^T \begin{pmatrix} I_m & 0 & 0 & \frac{1}{2}I_m \\ 0 & I_m & 0 & \frac{1}{2}I_m \\ 0 & 0 & I_m & \frac{1}{2}I_m \\ \frac{1}{2}I_m & \frac{1}{2}I_m & \frac{1}{2}I_m & I_m \end{pmatrix} \begin{pmatrix} \sqrt{\beta}A_2(x_2^k - \tilde{x}_2^k) \\ \sqrt{\beta}A_3(x_3^k - \tilde{x}_3^k) \\ \sqrt{\beta}A_4(x_4^k - \tilde{x}_4^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}.
\end{aligned}$$



Note that the eigenvalues of the symmetric matrix

$$\begin{pmatrix} 1 & 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix} \quad \text{are} \quad 1,\ 1 \text{ and } 1 \pm \tfrac{\sqrt{3}}{2},$$

and the smallest eigenvalue is $\frac{2-\sqrt{3}}{2}$. Therefore,

$$\begin{aligned}
\varphi(v^k, \tilde{v}^k) &\geq \frac{2-\sqrt{3}}{2} \begin{pmatrix} \sqrt{\beta}A_2(x_2^k - \tilde{x}_2^k) \\ \sqrt{\beta}A_3(x_3^k - \tilde{x}_3^k) \\ \sqrt{\beta}A_4(x_4^k - \tilde{x}_4^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}^T \begin{pmatrix} \sqrt{\beta}A_2(x_2^k - \tilde{x}_2^k) \\ \sqrt{\beta}A_3(x_3^k - \tilde{x}_3^k) \\ \sqrt{\beta}A_4(x_4^k - \tilde{x}_4^k) \\ \sqrt{1/\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix} \\
&= \frac{2-\sqrt{3}}{2} \|v^k - \tilde{v}^k\|_H^2,
\end{aligned}$$

and the assertion is proved. $\Box$

**Correction** Based on the predictor by (4.5), update the new iterate $v^{k+1}$ by

Update the new iterate $v^{k+1}$ by

$$v^{k+1} = v^k - \alpha_k (v^k - \tilde{v}^k), \qquad \alpha_k = \gamma \alpha_k^*, \qquad \gamma \in (0, 2) \tag{4.13a}$$

where

$$\alpha_k^* = \frac{\varphi(v^k, \tilde{v}^k)}{\|v^k - \tilde{v}^k\|_H^2} \tag{4.13b}$$

By using (4.10) and (4.13), we obtain

$$\|v^{k+1} - v^*\|_H^2 \leq \|v^k - v^*\|_H^2 - \gamma(2-\gamma)\alpha_k^* \varphi(v^k, \tilde{v}^k).$$

**Convergence** Using (4.12) and (4.13b), we obtain

$$\|v^{k+1} - v^*\|_H^2 \leq \|v^k - v^*\|_H^2 - \frac{\gamma(2-\gamma)(7-4\sqrt{3})}{4} \|v^k - \tilde{v}^k\|_H^2. \tag{4.14}$$

This inequality (4.14) is essential for the convergence.



# 5 Conclusion Remarks

All the methods in this note are prediction-correction methods. The first ADMM-based prediction-correction method was published by Ye and Yuan [13]. For solving the problem (1.1), we suggest to use the method in Section 3, whose prediction and correction is (3.1) and (3.12), respectively. Even though it needs to compute the step size in the correction step in each iteration, the cost is usually small in comparison with the computational load for solving the subproblem in the prediction step.

Since 2012, we have some new publication about the multi-block problems. First, without correction, the direct extension of the ADMM for the problem (1.1) is not necessarily convergent [2]. For multi-block problems, we suggest to use the methods [7, 8, 9, 11].

# References


[1] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Nov. 2010.

[2] C.H. Chen, B.S. He, Y.Y. Ye and X. M. Yuan, The direct extension of ADMM for multi-block convex minimization problems is not necessary convergent, Mathematical Programming, 155 (2016) 57-79.

[3] D. Gabay, "Applications of the Method of Multipliers to Variational Inequalities," in *Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems*, M. Fortin and R. Glowinski, eds., North Holland, Amsterdam, The Netherlands, 299-331, 1983.

[4] B.S. He, A class of projection and contraction methods for monotone variational inequalities, Applied Mathematics and Optimization, **35**, pp. 69-76, 1997.

[5] B.S. He, "Inexact implicit methods for monotone general variational inequalities", Math. Program. Vol. 86, Series A, pp. 199-217, 1999

[6] B.S. He, Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities, Computational Optimization and Applications, 42, 195-212, 2009.

[7] B.S. He, L.S. Hou, and X.M. Yuan, On Full Jacobian Decomposition of the Augmented Lagrangian Method for Separable Convex Programming, SIAM J. Optim., 25 (2015) 2274 – 2312.

[8] B.S. He, M. Tao and X.M. Yuan, Alternating Direction Method with Gaussian Back Substitution for Separable Convex Programming, SIAM J. Optim. 22(2012), 313-340.

[9] B.S. He, M. Tao and X.M. Yuan, A splitting method for separable convex programming, IMA J. Numerical Analysis, 31(2015), 394-426.

[10] B. S. He and M.-H. Xu, A general framework of contraction methods for monotone variational inequalities, Pacific J. Optimization, 4, 195-212, 2008.





[11] B.S. He and X.M. Yuan, Block-wise Alternating Direction Method of Multipliers for Multiple-block Convex Programming and Beyond, SMAI J. Computational Mathematics 1 (2015) 145-174.

[12] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Control Optim., 14 (1976), pp. 877-898.

[13] C. H. Ye and X. M. Yuan, A Descent Method for Structured Monotone Variational Inequalities, Optimization Methods and Software, Vol. 22, pp. 329-338, 2007.

# 凸优化和单调变分不等式的收缩算法

## 第十六讲: 三个可分离算子凸优化 的略有改动的交替方向法

## A slightly changed ADMM for convex optimization with three separable operators

南京大学数学系　　何炳生
hebma@nju.edu.cn

The context of this lecture is based on the publication [10] and [13]



**Abstract.** The classical alternating direction method of multipliers (ADMM) has been well studied in the context of linearly constrained convex programming and variational inequalities where the involved operator is formed as the sum of two individual functions without crossed variables. Recently, ADMM has found many novel applications in diversified areas such as image processing and statistics. However, it is still not clear whether ADMM can be extended to the case where the operator is the sum of more than two individual functions. In this lecture, we present a little changed ADMM for solving the linearly constrained separable convex optimization whose involved operator is separable into three individual functions. The $\mathcal{O}(1/t)$ convergence rate of the proposed methods is demonstrated.

**Keywords**: Alternating direction method, convex programming, linear constraint, separable structure, contraction method

---

## 1 Introduction

An important case of structured convex optimization problem is

$$\min\{\theta_1(x) + \theta_2(y) \mid Ax + y = b,\ x \in \mathcal{X},\ y \in \mathcal{Y}\}, \tag{1.1}$$

where $\theta_1 : \Re^n \to \Re$ and $\theta_2 : \Re^m \to \Re$ are closed proper convex functions (not necessarily smooth); $A \in \Re^{m\times n}$; $\mathcal{X} \subseteq \Re^n$ and $\mathcal{Y} \subseteq \Re^m$ are closed convex sets. The alternating direction method of multipliers (ADMM), which dates back to [6] and is closely related to the Douglas-Rachford operator splitting method [2], is perhaps the most popular method for solving (1.1). More specifically, for given $(y^k, \lambda^k)$ in the $k$-th iteration, it produces the new iterate in the following order:

$$\begin{cases} x^{k+1} = \text{Argmin}\big\{\theta_1(x) - (\lambda^k)^T Ax + \frac{\beta}{2}\|Ax + y^k - b\|^2 \;\big|\; x \in \mathcal{X}\big\}; \\ y^{k+1} = \text{Argmin}\big\{\theta_2(y) - (\lambda^k)^T y + \frac{\beta}{2}\|Ax^{k+1} + y - b\|^2 \big|\; y \in \mathcal{Y}\big\}; \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + y^{k+1} - b). \end{cases} \tag{1.2}$$

Therefore, ADMM can be viewed as a practical and structured-exploiting variant (split form or relaxed form) of ALM for solving the separable problem (1.1), with the adaption of minimizing the involved separable variables $x$ and $y$ separably in an alternating order. In fact, the iteration (1.2) is from $(y^k, \lambda^k)$ to $(y^{k+1}, \lambda^{k+1})$, $x$ is only an auxiliary variable in the iterative process. The sequence $\{(y^k, \lambda^k)\}$ generated by the recursion (1.2) satisfies



(see Theorem 1 in [12] by setting fixed $\beta$ and $\gamma \equiv 1$)

$$\begin{aligned} &\|\beta(y^{k+1} - y^*)\|^2 + \|\lambda^{k+1} - \lambda^*\|^2 \\ &\quad\le \quad \|\beta(y^k - y^*)\|^2 + \|\lambda^k - \lambda^*\|^2 - \big(\|\beta(y^k - y^{k+1})\|^2 + \|\lambda^k - \lambda^{k+1}\|^2\big). \end{aligned}$$

Because of its efficiency and easy implementation, ADMM has attracted wide attention of many authors in various areas, see e.g. [1, 7]. In particular, some novel and attractive applications of ADMM have been discovered very recently, e.g. the total-variation problem in image processing, the covariance selection problem and semidefinite least square problem in statistics [11], the semidefinite programming problems， the sparse and low-rank recovery problem in Engineering [14], and the matrix completion problem [1].

In some practical applications [4], the model is slightly more complicated than (1.1). The mathematical form of the problem is

$$\min\{\theta_1(x) + \theta_2(y) + \theta_3(z) | Ax + y + z = b,\ x \in \mathcal{X}, y \in \mathcal{Y}, z \in \mathcal{Z}\}, \tag{1.3}$$

where $\theta_1 : \Re^n \to \Re$, $\theta_2, \theta_3 : \Re^m \to \Re$ are closed proper convex functions (not necessarily smooth); $A \in \Re^{m\times n}$; $\mathcal{X} \subseteq \Re^n$, $\mathcal{Y}, \mathcal{Z} \subseteq \Re^m$ are closed convex sets. It is then natural to manage to extend ADMM to solve the problem (1.3), resulting in the following scheme:

$$\begin{cases} x^{k+1} = \text{Argmin}\big\{\theta_1(x) - (\lambda^k)^T Ax + \frac{\beta}{2}\|Ax + y^k + z^k - b\|^2 \;\big|\; x \in \mathcal{X}\big\}; \\ y^{k+1} = \text{Argmin}\big\{\theta_2(y) - (\lambda^k)^T y \;+\; \frac{\beta}{2}\|Ax^{k+1} + y + z^k - b\|^2 \;\big|\; y \in \mathcal{Y}\big\}; \\ z^{k+1} = \text{Argmin}\big\{\theta_3(z) - (\lambda^k)^T z \;+\; \frac{\beta}{2}\|Ax^{k+1} + y^{k+1} + z - b\|^2 \;\big|\; z \in \mathcal{Z}\big\}; \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + y^{k+1} + z^{k+1} - b), \end{cases} \tag{1.4}$$

and the involved subproblems of (1.4) are solved consecutively in the ADMM manner. Unfortunately, with the $(y^{k+1}, z^{k+1}, \lambda^{k+1})$ offered by (1.4), the convergence of the extended ADMM (1.4) is still open.

In this paper, we present a little changed alternating direction method for the problem (1.3). Again, based on $(y^{k+1}, z^{k+1}, \lambda^{k+1})$ offered by (1.4), we set

$$(y^{k+1}, z^{k+1}, \lambda^{k+1}) := (y^{k+1} + (z^k - z^{k+1}), z^{k+1}, \lambda^{k+1}). \tag{1.5}$$

Note that the change of (1.5) is small. In addition, for the problem with two separable operators, by setting $z^k = 0$ for all $k$, the proposed method is just reduced to the algorithm (1.2) for the problem (1.1). Therefore, we call the proposed method *a little*



*changed alternating direction method of multipliers for convex optimization with three separable operators*.

The outline of this paper is as follows. In Section 2, we convert the problem (1.3) to the equivalent variational inequality and characterize its solution set. Section 3 shows the contraction property of the proposed method. In Section 4, we define an auxiliary vector and derive its main associated properties, and show the $\mathcal{O}(1/t)$ convergence rate of the proposed method. Finally, some conclusions are made in Section 6.

## 2 The variational inequality characterization

Throughout, we assume that the solution set of (1.3) is not empty. The convergence analysis is based on the tool of variational inequality. For this purpose, we define

$$\mathcal{W} = \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \times \Re^m.$$

It is easy to verify that the convex programming problem (1.3) is characterized by the following variational inequality: Find $w^* = (x^*, y^*, z^*, \lambda^*) \in \mathcal{W}$ such that

$$\left\{\begin{array}{l} \theta_1(x) - \theta_1(x^*) + (x - x^*)^T(-A^T\lambda^*) \geq 0, \\ \theta_2(y) - \theta_2(y^*) + (y - y^*)^T(-\lambda^*) \geq 0, \\ \theta_3(z) - \theta_3(z^*) + (z - z^*)^T(-\lambda^*) \geq 0, \\ \qquad\qquad (\lambda - \lambda^*)^T(Ax^* + y^* + z^* - b) \geq 0, \end{array}\right. \quad \forall\, w \in \mathcal{W}, \tag{2.1}$$

or in the more compact form:

$$\text{VI}(\mathcal{W}, F, \theta) \qquad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall\, w \in \mathcal{W}, \tag{2.2}$$

where

$$\theta(u) = \theta_1(x) + \theta_2(y) + \theta_3(z),$$



and

$$u = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \qquad w = \begin{pmatrix} x \\ y \\ z \\ \lambda \end{pmatrix}, \qquad F(w) = \begin{pmatrix} -A^T\lambda \\ -\lambda \\ -\lambda \\ Ax + y + z - b \end{pmatrix}. \tag{2.3}$$

Note that $F(w)$ defined in (2.3) is monotone. Under the nonempty assumption on the solution set of (1.3), the solution set of (2.2)-(2.3), denoted by $\mathcal{W}^*$, is also nonempty.

The Theorem 2.3.5 in [5] provides an insightful characterization for the solution set of a generic VI. This characterization actually provides us a novel and simple approach which enables us to derive the $O(1/t)$ convergence rate for the original ADMM in [13]. In the following theorem, we specify this result for the derived VI$(\mathcal{W}, F, \theta)$. Note that the proof of the next theorem is an incremental extension of Theorem 2.3.5 in [5] and also Theorem 2.1 in [13]. But, we include all the details because of its crucial importance in our analysis.

**Theorem 2.1** *The solution set of VI*$(\mathcal{W}, F, \theta)$ *is convex and it can be characterized as*

$$\mathcal{W}^* = \bigcap_{w \in \mathcal{W}} \big\{\bar{w} \in \mathcal{W} : \big(\theta(u) - \theta(\bar{u})\big) + (w - \bar{w})^T F(w) \geq 0\big\}. \tag{2.4}$$

**Proof**. Indeed, if $\bar{w} \in \mathcal{W}^*$, according to (2.2) we have

$$\theta(u) - \theta(\bar{u}) + (w - \bar{w})^T F(\bar{w}) \geq 0, \quad \forall\, w \in \mathcal{W}.$$

By using the monotonicity of $F$ on $\mathcal{W}$, this implies

$$\theta(u) - \theta(\bar{u}) + (w - \bar{w})^T F(w) \geq 0, \quad \forall\, w \in \mathcal{W}.$$

Thus, $\bar{w}$ belongs to the right-hand set in (2.4).

Conversely, suppose $\bar{w}$ belongs to the latter set. Let $w \in \mathcal{W}$ be arbitrary. The vector

$$\tilde{w} = \tau\bar{w} + (1 - \tau)w$$

belongs to $\mathcal{W}$ for all $\tau \in (0, 1)$. Thus we have

$$\theta(\tilde{u}) - \theta(\bar{u}) + (\tilde{w} - \bar{w})^T F(\tilde{w}) \geq 0. \tag{2.5}$$



Because $\theta(\cdot)$ is convex and $\tilde{u} = \tau\bar{u} + (1 - \tau)u$, we have

$$\theta(\tilde{u}) \leq \tau\theta(\bar{u}) + (1 - \tau)\theta(u).$$

Substituting it in (2.5), we get

$$(\theta(u) - \theta(\bar{u})) + (w - \bar{w})^T F(\tau\bar{w} + (1 - \tau)w) \geq 0$$

for all $\tau \in (0, 1)$. Letting $\tau \to 1$ yields

$$(\theta(u) - \theta(\bar{u})) + (w - \bar{w})^T F(\bar{w}) \geq 0.$$

Thus $\bar{w} \in \mathcal{W}^*$. Now, we turn to prove the convexity of $\mathcal{W}^*$. For each fixed but arbitrary $w \in \mathcal{W}$, the set

$$\{\bar{w} \in \mathcal{W} :\ \theta(\bar{u}) + \bar{w}^T F(w) \leq \theta(u) + w^T F(w)\}$$

is convex and so is the equivalent set

$$\{\bar{w} \in \mathcal{W} :\ \big(\theta(u) - \theta(\bar{u})\big) + (w - \bar{w})^T F(w) \geq 0\}.$$

Since the intersection of any number of convex sets is convex, it follows that the solution set of VI$(\mathcal{W}, F, \theta)$ is convex. $\square$

Theorem 2.1 thus implies that $\bar{w} \in \mathcal{W}$ is an approximate solution of VI$(\mathcal{W}, F, \theta)$ with the accuracy $\epsilon > 0$ if it satisfies

$$\theta(u) - \theta(\bar{u}) + F(w)^T(w - \bar{w}) \geq -\epsilon, \;\; \forall\, w \in \mathcal{W}.$$

In this paper, we show that, for given $\epsilon > 0$ and a substantial compact set $\mathcal{D} \subset \mathcal{W}$, after $t$ iterations of the proposed methods, we can find a $\bar{w} \in \mathcal{W}$ such that

$$\hat{w} \in \mathcal{W} \quad \text{and} \quad \sup_{w \in \mathcal{D}} \big\{ \theta(\hat{u}) - \theta(u) + (\hat{w} - w)^T F(w) \big\} \leq \epsilon. \tag{2.6}$$

The convergence rate $\mathcal{O}(1/t)$ of the proposed methods is thus established.

For convenience of coming analysis, we define the following matrices:

$$P = \begin{pmatrix} \beta I_m & 0 & 0 \\ \beta I_m & \beta I_m & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix}, \qquad D = \begin{pmatrix} \beta I_m & 0 & 0 \\ 0 & \beta I_m & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix}. \tag{2.7}$$



Note that for the above defined matrices $M$ and $D$, we have

$$D^{-1}P = \begin{pmatrix} I_m & 0 & 0 \\ I_m & I_m & 0 \\ 0 & 0 & I_m \end{pmatrix}, \quad P^{-T}D = \begin{pmatrix} I_m & -I_m & 0 \\ 0 & I_m & 0 \\ 0 & 0 & I_m \end{pmatrix}. \tag{2.8}$$

# 3 Contraction property of the proposed method

In the alternating direction method, $x$ is only the auxiliary variable in the iteration process. For convenience of analysis, we use the notation

$$v = (y, z, \lambda),$$

which is a sub-vector of $w$. For $w^* \in \mathcal{W}^*$, we also define

$$\mathcal{V}^* := \{ v^* = (y^*, z^*, \lambda^*) \,|\, (x^*, y^*, z^*, \lambda^*) \in \mathcal{W}^* \}.$$

In addition, we divide each iteration of the proposed method in two steps-the prediction step and the correction step. From a given $v^k = (y^k, z^k, \lambda^k)$, we use

$\bar{w}^k = (\bar{x}^k, \bar{y}^k, \bar{z}^k, \bar{\lambda}^k) \in \mathcal{W}$ to denote the solution of (1.4) and call it as the prediction step.

**The prediction step in the $k$-th iteration (use ADMM manner):**
Begin with given $v^k = (y^k, z^k, \lambda^k)$, generate $\bar{w}^k = (\bar{x}^k, \bar{y}^k, \bar{z}^k, \bar{\lambda}^k)$ in the following order:

$$\bar{x}^k = \text{Argmin}\big\{\theta_1(x) + \frac{\beta}{2}\|(Ax + y^k + z^k - b) - \frac{1}{\beta}\lambda^k\|^2 \mid x \in \mathcal{X}\big\}, \tag{3.1a}$$

$$\bar{y}^k = \text{Argmin}\big\{\theta_2(y) + \frac{\beta}{2}\|(A\bar{x}^k + y + z^k - b) - \frac{1}{\beta}\lambda^k\|^2 \mid y \in \mathcal{Y}\big\}, \tag{3.1b}$$

$$\bar{z}^k = \text{Argmin}\big\{\theta_3(z) + \frac{\beta}{2}\|(A\bar{x}^k + \bar{y}^k + z - b) - \frac{1}{\beta}\lambda^k\|^2 \mid z \in \mathcal{Z}\big\}, \tag{3.1c}$$

$$\bar{\lambda}^k = \lambda^k - \beta(A\bar{x}^k + \bar{y}^k + \bar{z}^k - b). \tag{3.1d}$$

For the new iterative loop, we need only to produce $v^{k+1} = (y^{k+1}, z^{k+1}, \lambda^{k+1})$. Use the notation of $\bar{w}^k$, the update form (1.5) can be written as



**The correction step**: Update the new iterate $v^{k+1} = (y^{k+1}, z^{k+1}, \lambda^{k+1})$ by

$$\begin{pmatrix} y^{k+1} \\ z^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ z^k \\ \lambda^k \end{pmatrix} - \begin{pmatrix} I_m & -I_m & 0 \\ 0 & I_m & 0 \\ 0 & 0 & I_m \end{pmatrix} \begin{pmatrix} y^k - \bar{y}^k \\ z^k - \bar{z}^k \\ \lambda^k - \bar{\lambda}^k \end{pmatrix}. \tag{3.2}$$

Using the notation $P^{-T}D$, the correction step can be written as

$$v^{k+1} = v^k - P^{-T}D(v^k - \bar{v}^k).$$

We consider the general correction update form

$$v^{k+1} = v^k - \alpha P^{-T}D(v^k - \bar{v}^k), \quad \alpha \in (0, 1]. \tag{3.3}$$

In other words, the update form (3.2) is a special case of (3.3) with $\alpha = 1$. Taking $\alpha \in (0, 1)$, the method is a special case of the method proposed in [10].

## 3.1 Properties of the vector $\bar{w}^k$ by the prediction step

We establish the following lemma.

**Lemma 3.1** *Let* $\bar{w}^k = (\bar{x}^k, \bar{y}^k, \bar{z}^k, \bar{\lambda}^k)$ *be generated by the alternating direction-prediction step* (3.1a)–(3.1d) *from the given vector* $v^k = (y^k, z^k, \lambda^k)$*. Then, we have* $\bar{w}^k \in \mathcal{W}$ *and*

$$\theta(u) - \theta(\bar{u}^k) + (w - \bar{w}^k)^T d(v^k, \bar{w}^k) \geq (v - \bar{v}^k)^T P(v^k - \bar{v}^k), \ \forall\, w \in \mathcal{W}, \tag{3.4}$$

*where*

$$d(v^k, \bar{w}^k) = F(\bar{w}^k) + \eta(v^k, \bar{v}^k), \tag{3.5}$$

$$\eta(v^k, \bar{v}^k) = \begin{pmatrix} A^T \\ I_m \\ I_m \\ 0 \end{pmatrix} \beta\big((y^k - \bar{y}^k) + (z^k - \bar{z}^k)\big), \tag{3.6}$$

*and the matrix* $P$ *is defined in (2.7).*



**Proof**. The proof consists of some manipulations. Recall (3.1a)–(3.1c). We have $\bar{u}^k \in \mathcal{U}$ and

$$\begin{cases} \theta_1(x) - \theta_1(\bar{x}^k) + (x - \bar{x}^k)^T (A^T[\beta(A\bar{x}^k + y^k + z^k - b) - \lambda^k]) \geq 0, \\ \theta_2(y) - \theta_2(\bar{y}^k) + (y - \bar{y}^k)^T (\beta(A\bar{x}^k + \bar{y}^k + z^k - b) - \lambda^k) \geq 0, \\ \theta_3(z) - \theta_3(\bar{z}^k) + (z - \bar{z}^k)^T (\beta(A\bar{x}^k + \bar{y}^k + \bar{z}^k - b) - \lambda^k) \geq 0, \end{cases}$$

for all $u \in \mathcal{U}$. Using (3.1d), the above inequality can be written as

$$\theta(u) - \theta(\bar{u}^k) + \begin{pmatrix} x - \bar{x}^k \\ y - \bar{y}^k \\ z - \bar{z}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T \bar{\lambda}^k \\ -\bar{\lambda}^k \\ -\bar{\lambda}^k \end{pmatrix} + \begin{pmatrix} \beta A^T (y^k - \bar{y}^k) + \beta A^T (z^k - \bar{z}^k) \\ \beta(z^k - \bar{z}^k) \\ 0 \end{pmatrix} \right\} \geq 0. \tag{3.7}$$

Adding the following term

$$\begin{pmatrix} x - \bar{x}^k \\ y - \bar{y}^k \\ z - \bar{z}^k \end{pmatrix}^T \begin{pmatrix} 0 \\ \beta(y^k - \bar{y}^k) \\ \beta(y^k - \bar{y}^k) + \beta(z^k - \bar{z}^k) \end{pmatrix}$$

to the both sides of the last inequality, we get $\bar{w}^k \in \mathcal{W}$ and

$$\theta(u)-\theta(\bar{u}^k)+\begin{pmatrix} x-\bar{x}^k \\ y-\bar{y}^k \\ z-\bar{z}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\bar{\lambda}^k \\ -\bar{\lambda}^k \\ -\bar{\lambda}^k \end{pmatrix} + \begin{pmatrix} A^T\beta\big((y^k-\bar{y}^k)+(z^k-\bar{z}^k)\big) \\ \beta\big((y^k-\bar{y}^k)+(z^k-\bar{z}^k)\big) \\ \beta\big((y^k-\bar{y}^k)+(z^k-\bar{z}^k)\big) \end{pmatrix} \right\}$$

$$\geq \begin{pmatrix} x-\bar{x}^k \\ y-\bar{y}^k \\ z-\bar{z}^k \end{pmatrix}^T \begin{pmatrix} 0 \\ \beta(y^k-\bar{y}^k) \\ \beta(y^k-\bar{y}^k)+\beta(z^k-\bar{z}^k) \end{pmatrix}, \quad \forall\, u \in \mathcal{U}. \tag{3.8}$$

Because

$$A\bar{x}^k+\bar{y}^k+\bar{z}^k-b = \frac{1}{\beta}(\lambda^k-\bar{\lambda}^k),$$

adding the equal terms

$$(\lambda-\bar{\lambda}^k)^T(A\bar{x}^k+\bar{y}^k+\bar{z}^k-b) \quad \text{and} \quad (\lambda-\bar{\lambda}^k)^T\frac{1}{\beta}(\lambda^k-\bar{\lambda}^k)$$



to the left side and right side of (3.8), respectively, we get $\bar{w}^k \in \mathcal{W}$ and

$$\theta(u)-\theta(\bar{u}^k)+\begin{pmatrix} x-\bar{x}^k \\ y-\bar{y}^k \\ z-\bar{z}^k \\ \lambda-\bar{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\bar{\lambda}^k \\ -\bar{\lambda}^k \\ -\bar{\lambda}^k \\ A\bar{x}^k+\bar{y}^k+\bar{z}^k-b \end{pmatrix} + \begin{pmatrix} A^T\beta\big((y^k-\bar{y}^k)+(z^k-\bar{z}^k)\big) \\ \beta\big((y^k-\bar{y}^k)+(z^k-\bar{z}^k)\big) \\ \beta\big((y^k-\bar{y}^k)+(z^k-\bar{z}^k)\big) \\ 0 \end{pmatrix} \right\}$$

$$\geq \begin{pmatrix} y-\bar{y}^k \\ z-\bar{z}^k \\ \lambda-\bar{\lambda}^k \end{pmatrix}^T \begin{pmatrix} \beta(y^k-\bar{y}^k) \\ \beta(y^k-\bar{y}^k)+\beta(z^k-\bar{z}^k) \\ \frac{1}{\beta}(\lambda^k-\bar{\lambda}^k) \end{pmatrix}, \quad \forall\, w \in \mathcal{W}.$$

Using the notations of $F(w)$, $d(v^k,\bar{w}^k)$ and $P$, the assertion follows immediately and the lemma is proved. $\square$

**Lemma 3.2** *Let $\bar{w}^k=(\bar{x}^k,\bar{y}^k,\bar{z}^k,\bar{\lambda}^k)$ be generated by the alternating direction prediction step (3.1a)–(3.1d) from the given vector $v^k=(y^k,z^k,\lambda^k)$. Then for all*

*$v^* \in \mathcal{V}^*$, we have*

$$
\begin{aligned}
&(v^k - v^*)^T P(v^k - \bar{v}^k) \\
&\quad \geq \quad \frac{1}{2}\|v^k - \bar{v}^k\|_D^2 + \frac{1}{2}\beta\|(y^k - \bar{y}^k) + (z^k - \bar{z}^k) + \frac{1}{\beta}(\lambda^k - \bar{\lambda}^k)\|^2, \quad (3.9)
\end{aligned}
$$

*where the matrices $P$ is defined in (2.7).*

**Proof**. Setting $w = w^*$ in (3.4) (notice that $v^*$ is a sub-vector of $w^*$), it yields that

$$
\begin{aligned}
(\bar{v}^k - v^*)^T P(v^k - \bar{v}^k) \quad \geq \quad & \big(\theta(\bar{u}^k) - \theta(u^*)\big) + (\bar{w}^k - w^*)^T F(\bar{w}^k) \\
& + (\bar{w}^k - w^*)^T \eta(v^k, \bar{v}^k). \qquad (3.10)
\end{aligned}
$$

Now, we deal with the last term in the right hand side of the inequality (3.10). By using the notation of $\eta(v^k, \bar{v}^k)$, $Ax^* + y^* + z^* = b$ and (3.1d), we obtain

$$
\begin{aligned}
&(\bar{w}^k - w^*)^T \eta(v^k, \bar{v}^k) \\
&\quad = \quad \beta(A\bar{x}^k + \bar{y}^k + \bar{z}^k - Ax^* - y^* - z^*)^T \{(y^k - \bar{y}^k) + (z^k - \bar{z}^k)\} \\
&\quad = \quad \beta(A\bar{x}^k + \bar{y}^k + \bar{z}^k - b)^T \{(y^k - \bar{y}^k) + (z^k - \bar{z}^k)\} \\
&\quad = \quad (\lambda^k - \bar{\lambda}^k)^T \{(y^k - \bar{y}^k) + (z^k - \bar{z}^k)\}.
\end{aligned}
$$



Substituting it in (3.10), we get

$$
\begin{aligned}
(\bar{v}^k - v^*)^T P(v^k - \bar{v}^k) \ \geq \ & \big(\theta(\bar{u}^k) - \theta(u^*)\big) + (\bar{w}^k - w^*)^T F(\bar{w}^k) \\
& + (\lambda^k - \bar{\lambda}^k)^T \{(y^k - \bar{y}^k) + (z^k - \bar{z}^k)\}. \quad (3.11)
\end{aligned}
$$

Since $F$ is monotone, we have

$$
\theta(\bar{u}^k) - \theta(u^*) + (\bar{w}^k - w^*)^T F(\bar{w}^k) \geq \theta(\bar{u}^k) - \theta(u^*) + (\bar{w}^k - w^*)^T F(w^*) \geq 0.
$$

Substituting it in the right hand side of (3.11), we obtain

$$
(\bar{v}^k - v^*)^T P(v^k - \bar{v}^k) \geq (\lambda^k - \bar{\lambda}^k)^T \{(y^k - \bar{y}^k) + (z^k - \bar{z}^k)\}.
$$

It follows from the last equality that

$$
\begin{aligned}
&(v^k - v^*)^T P(v^k - \bar{v}^k) \\
&\quad \geq (v^k - \bar{v}^k)^T P(v^k - \bar{v}^k) + (\lambda^k - \bar{\lambda}^k)^T \{(y^k - \bar{y}^k) + (z^k - \bar{z}^k)\}. \quad (3.12)
\end{aligned}
$$

Observe the matrices $P$ and $D$ (see (2.7)), by a manipulation, the right hand side of (3.12) can be written as

$$\begin{aligned}
&(v^k-\bar v^k)^T P(v^k-\bar v^k) + (\lambda^k-\bar\lambda^k)^T\{(y^k-\bar y^k)+(z^k-\bar z^k)\}\\
= &\begin{pmatrix} y^k-\bar y^k \\ z^k-\bar z^k \\ \lambda^k-\bar\lambda^k \end{pmatrix}^T \begin{pmatrix} \beta I_m & \frac12\beta I_m & \frac12 I_m \\ \frac12\beta I_m & \beta I_m & \frac12 I_m \\ \frac12 I_m & \frac12 I_m & \frac1\beta I_m \end{pmatrix} \begin{pmatrix} y^k-\bar y^k \\ z^k-\bar z^k \\ \lambda^k-\bar\lambda^k \end{pmatrix}\\
= &\frac12\|v^k-\bar v^k\|_D^2 + \frac12\beta\|(y^k-\bar y^k)+(z^k-\bar z^k)+\frac1\beta(\lambda^k-\bar\lambda^k)\|^2.
\end{aligned}$$

Substituting it in the right hand side of (3.12), the assertion of this lemma is proved. □

Whenever $v^k\neq\bar v^k$, the right hand side of (3.9) is positive. For any positive definite matrix $H$, (3.9) implies that

$$\begin{aligned}
&\langle H(v^k-v^*), H^{-1}P(v^k-\bar v^k)\rangle\\
&\quad\ge \frac12\|v^k-\bar v^k\|_D^2 + \frac12\beta\|(y^k-\bar y^k)+(z^k-\bar z^k)+\frac1\beta(\lambda^k-\bar\lambda^k)\|^2,
\end{aligned}$$



and $H^{-1}P(v^k-\bar v^k)$ is an ascent direction of the distance function $\frac12\|v-v^*\|_H^2$ at the point $v^k$. By choosing

$$H = PD^{-1}P^T = \begin{pmatrix} \beta I_m & \beta I_m & 0 \\ \beta I_m & 2\beta I_m & 0 \\ 0 & 0 & \frac1\beta I_m \end{pmatrix}, \tag{3.13}$$

and using the matrix $P^{-T}D$ (see (2.8)), we have

$$H^{-1}P = P^{-T}D = \begin{pmatrix} I_m & -I_m & 0 \\ 0 & I_m & 0 \\ 0 & 0 & I_m \end{pmatrix}.$$

## 3.2 Correction and the contractive property

The alternating direction-prediction step begins with given $v^k=(y^k,z^k,\lambda^k)$. In fact $x$ is only the auxiliary variable in the iteration process. For the new loop, we need only to produce $v^{k+1}=(y^{k+1},z^{k+1},\lambda^{k+1})$ and we call this step as the **correction step**.

**Theorem 3.1** *Let $\bar w^k=(\bar x^k,\bar y^k,\bar z^k,\bar\lambda^k)$ be generated by the alternating* *direction-prediction step* (3.1a)–(3.1d) *from the given vector* $v^k = (y^k, z^k, \lambda^k)$ *and* $v^{k+1}$ *be given by* (3.3). *Then we have*

$$\begin{aligned} \|v^{k+1} - v^*\|_H^2 &\le \|v^k - v^*\|_H^2 - \alpha(1-\alpha)\|v^k - \bar{v}^k\|_D^2 \\ &\quad -\alpha\beta\|(y^k - \bar{y}^k) + (z^k - \bar{z}^k) + \frac{1}{\beta}(\lambda^k - \bar{\lambda}^k)\|^2, \quad \forall v^* \in \mathcal{V}^*, \end{aligned} \tag{3.14}$$

*where* $H$ *is defined in* (3.13).

**Proof**. By using (3.3) and the definition of the matrix $H$, we have

$$\begin{aligned} &\|v^k - v^*\|_H^2 - \|v^{k+1} - v^*\|_H^2 \\ &= \|v^k - v^*\|_H^2 - \|(v^k - v^*) - (v^k - v^{k+1})\|_H^2 \\ &= \|v^k - v^*\|_H^2 - \|(v^k - v^*) - \alpha P^{-T} D(v^k - \bar{v}^k)\|_H^2 \\ &= 2\alpha(v^k - v^*)^T P(v^k - \bar{v}^k) - \alpha^2\|v^k - \bar{v}^k\|_D^2. \end{aligned}$$

Substituting (3.9) in the last equality, we get

$$\begin{aligned} &\|v^k - v^*\|_H^2 - \|v^{k+1} - v^*\|_H^2 \\ &\ge \alpha(1-\alpha)\|v^k - \bar{v}^k\|_D^2 + \alpha\beta\|(y^k - \bar{y}^k) + (z^k - \bar{z}^k) + \frac{1}{\beta}(\lambda^k - \bar{\lambda}^k)\|^2. \end{aligned}$$



The assertion of this theorem is proved. $\square$

For any $\alpha \in (0,1)$, it follows (3.14) that

$$\|v^{k+1} - v^*\|_H^2 \le \|v^k - v^*\|_H^2 - \alpha(1-\alpha)\|v^k - \bar{v}^k\|_D^2, \quad \forall\, v^* \in \mathcal{V}^*. \tag{3.15}$$

The above inequality is essential for the global convergence of the method using update form (3.3) with $\alpha \in (0,1)$, see [10]. In fact, based on (3.15), the sequence $\{v^k\}$ is bounded, we have

$$\lim_{k\to\infty} \|v^k - \bar{v}^k\|_D^2 = 0. \tag{3.16}$$

Consequently the sequence $\{\bar{v}^k\}$ is also bounded and it converges to a limit point $v^\infty$. On the other hand, due to Lemma 3.1, we have

$$\begin{aligned} &\theta(u) - \theta(\bar{u}^k) + (w - \bar{w}^k)^T F(\bar{w}^k) \\ &\ge -(w - \bar{w}^k)^T \eta(v^k, \bar{v}^k) + (v - \bar{v}^k)^T P(v^k - \bar{v}^k), \quad \forall\, w \in \mathcal{W}. \end{aligned} \tag{3.17}$$

From (3.16) and (3.17) we can derive $v^\infty \in \mathcal{V}^*$ and the induced $w^\infty$ is a solution of VI$(\mathcal{W}, F, \theta)$.

# 4 Convergence rate in an ergodic sense

The convergence in the last section is only for $\alpha \in (0, 1)$ with update form (3.3). Since (1.5), namely, the little changed alternating direction method of multipliers, is equivalent to (3.2) with $\alpha = 1$, it follows from Theorem 3.1 that

$$\|v^{k+1}-v^*\|_H^2 \le \|v^k-v^*\|_H^2-\beta\|(y^k-\bar{y}^k)+(z^k-\bar{z}^k)+\frac{1}{\beta}(\lambda^k-\bar{\lambda}^k)\|^2, \ \forall v^* \in \mathcal{V}^*.$$

The sequence $\{\|v^k - v^*\|_H\}$ is monotonically non-increasing. However, we have not obtained the global convergence in the contraction sense. This section, however, shows that the method using update form (3.2) has convergence rate $O(1/t)$ for all $\alpha \in (0, 1]$ in an ergodic sense.

According to Theorem 2.1, for given $\epsilon > 0$ and a substantial compact set $\mathcal{D} \subset \mathcal{W}$, our task is to find a $\tilde{w}$ such that (see (2.6))

$$\tilde{w} \in \mathcal{W} \quad \text{and} \quad \sup_{w \in \mathcal{D}} \{\theta(\tilde{u}) - \theta(u) + (\tilde{w} - w)^T F(w)\} \le \epsilon,$$

in $O(1/\epsilon)$ iterations. Generally, our complexity analysis follows the line of [3, 13], but instead of using $\bar{w}^k$ directly, we need first to introduce an auxiliary vector.



**The additional auxiliary vector**

$$\tilde{w}^k = \begin{pmatrix} \tilde{x}^k \\ \tilde{y}^k \\ \tilde{z}^k \\ \tilde{\lambda}^k \end{pmatrix}, \quad \text{where} \quad \tilde{u}^k = \bar{u}^k \tag{4.1a}$$

and

$$\tilde{\lambda}^k = \bar{\lambda}^k - \beta\big((y^k - \bar{y}^k) + (z^k - \bar{z}^k)\big). \tag{4.1b}$$

In order to rewrite the assertion in Lemma 3.1 in form of $\tilde{w}$, we need the following lemmas to express the terms $d(v^k, \bar{w}^k)$ and $P(v^k - \bar{v}^k)$ in form of $w^k$ and $\tilde{w}^k$.

**Lemma 4.1** *For the $\tilde{w}^k$ defined in* (4.1) *and the $\bar{w}^k$ generated by* (3.1)*, we have*

$$d(v^k, \bar{w}^k) = F(\tilde{w}^k), \tag{4.2}$$

*where $d(v^k, \bar{w}^k)$ defined in* (3.5). *In addition, it holds that*

$$P(v^k - \bar{v}^k) = Q(v^k - \tilde{v}^k), \tag{4.3}$$

*where*

$$Q = \begin{pmatrix} \beta I_m & 0 & 0 \\ \beta I_m & \beta I_m & 0 \\ -I_m & -I_m & \frac{1}{\beta} I_m \end{pmatrix}. \tag{4.4}$$

**Proof**. Since $\tilde{u}^k = \bar{u}^k$ and $\tilde{\lambda}^k = \bar{\lambda}^k - \beta\big((y^k - \bar{y}^k) + (z^k - \bar{z}^k)\big)$, we have

$$\beta\big((y^k - \bar{y}^k) + (z^k - \bar{z}^k)\big) = \bar{\lambda}^k - \tilde{\lambda}^k. \tag{4.5}$$

Substituting it in the notation of $\eta(v^k, \bar{v}^k)$ (see (3.6)), we get

$$\begin{aligned} d(v^k, \bar{w}^k) &= F(\bar{w}^k) + \eta(v^k, \bar{v}^k) \\ &= \begin{pmatrix} -A^T\bar{\lambda}^k \\ -\bar{\lambda}^k \\ -\bar{\lambda}^k \\ A\bar{x}^k + \bar{y}^k + \bar{z}^k - b \end{pmatrix} + \begin{pmatrix} A^T \\ I \\ I \\ 0 \end{pmatrix} (\bar{\lambda}^k - \tilde{\lambda}^k) \;=\; F(\tilde{w}^k). \end{aligned}$$



The last equation is due to $(\bar{x}^k, \bar{y}^k, \bar{z}^k) = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k)$. The equation (4.1b) implies

$$\begin{aligned} \frac{1}{\beta}(\lambda^k - \bar{\lambda}^k) &= \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k) - \big((y^k - \bar{y}^k) + (z^k - \bar{z}^k)\big) \\ &= \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k) - \big((y^k - \tilde{y}^k) + (z^k - \tilde{z}^k)\big). \end{aligned}$$

Using the matrix $P$ (see (2.7)), (4.1a) and the last equation, we obtain

$$\begin{aligned} P(v^k - \bar{v}^k) &= \begin{pmatrix} \beta I_m & 0 & 0 \\ \beta I_m & \beta I_m & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} y^k - \bar{y}^k \\ z^k - \bar{z}^k \\ \lambda^k - \bar{\lambda}^k \end{pmatrix} \\ &= \begin{pmatrix} \beta I_m & 0 & 0 \\ \beta I_m & \beta I_m & 0 \\ -I_m & -I_m & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} = Q(v^k - \tilde{v}^k). \end{aligned}$$

Thus (4.3) holds and the lemma is proved. $\square$

By using (4.2), the assertion in Lemma 3.1 can be rewritten accordingly in the following lemma.

**Lemma 4.2** *Let* $\bar{w}^k = (\bar{x}^k, \bar{y}^k, \bar{z}^k, \bar{\lambda}^k)$ *be generated by the alternating direction-prediction step* (3.1a)–(3.1d) *from the given vector* $v^k = (y^k, z^k, \lambda^k)$ *and* $\tilde{w}^k$ *be defined by* (4.1). *Then we have* $\tilde{w}^k \in \mathcal{W}$ *and*

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \ge (v - \tilde{v}^k)^T P(v^k - \bar{v}^k), \ \forall w \in \mathcal{W}. \quad (4.6)$$

**Proof**. The assertion follows from (3.4), (4.2) and (4.1). □

Now, we are ready to prove the key inequalities for the convergence rate of the proposed method, which are given in the following lemmas.

**Lemma 4.3** *Let* $\bar{w}^k = (\bar{x}^k, \bar{y}^k, \bar{z}^k, \bar{\lambda}^k)$ *be generated by the alternating direction-prediction step* (3.1a)–(3.1d) *from the given vector* $v^k = (y^k, z^k, \lambda^k)$ *and* $\tilde{w}^k$ *be defined by* (4.1). *If the new iterate* $v^{k+1}$ *is updated by* (3.3), *then we have*

$$\begin{aligned} &\big(\theta(u) - \theta(\tilde{u}^k)\big) + (w - \tilde{w}^k)^T F(\tilde{w}^k) + \frac{1}{2}(\|v - v^k\|_H^2 - \|v - v^{k+1}\|_H^2) \\ &\quad \ge \ \frac{1}{2}(\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2), \quad \forall w \in \mathcal{W}, \end{aligned} \quad (4.7)$$

*where matrix* $H$ *is defined in* (3.13).



**Proof**. According to the update form (3.3), we have

$$v^k - \bar{v}^k = D^{-1} P^T (v^k - v^{k+1}).$$

Substituting it into the right hand side of (4.6) and using $PD^{-1}P^T = H$, we obtain

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \ge (v - \tilde{v}^k)^T H(v^k - v^{k+1}), \ \forall w \in \mathcal{W}. \quad (4.8)$$

By setting

$$a = v, \qquad b = \tilde{v}^k, \qquad c = v^k, \qquad \text{and} \qquad d = v^{k+1},$$

in the identity

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

we obtain

$$\begin{aligned} &(v - \tilde{v}^k)^T H(v^k - v^{k+1}) \\ &\quad = \ \frac{1}{2}(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2) + \frac{1}{2}(\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2). \end{aligned}$$

Substituting it in the right hand side of (4.8) and by a manipulation, we get (4.7) and the lemma is proved. □

**Lemma 4.4** *Let $\bar{w}^k = (\bar{x}^k, \bar{y}^k, \bar{z}^k, \bar{\lambda}^k)$ be generated by the alternating direction prediction step* (3.1a)–(3.1d) *from the given vector $v^k = (y^k, z^k, \lambda^k)$ and $\tilde{w}^k$ be defined by* (4.1)*. If the new iterate $v^{k+1}$ is updated by* (3.3)*, then we have*

$$\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2 = \frac{\alpha}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2 + \alpha(1-\alpha)\|v^k - \bar{v}^k\|_D^2. \quad (4.9)$$

*where matrix $H$ is defined in* (3.13)*.*

**Proof**. In view of the update form (3.3) and $H = PD^{-1}P^T$, we have

$$\begin{aligned}
&\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2 \\
&\quad = \|v^k - \tilde{v}^k\|_H^2 - \|v^k - \tilde{v}^k + (v^{k+1} - v^k)\|_H^2 \\
&\quad = \|v^k - \tilde{v}^k\|_H^2 - \|v^k - \tilde{v}^k - \alpha P^{-T} D(v^k - \bar{v}^k)\|_H^2 \\
&\quad = 2\alpha(v^k - \tilde{v}^k)^T P(v^k - \bar{v}^k) - \alpha^2\|P^{-T} D(v^k - \bar{v}^k)\|_H^2 \\
&\quad = 2\alpha(v^k - \tilde{v}^k)^T P(v^k - \bar{v}^k) - \alpha^2\|v^k - \bar{v}^k\|_D^2. \qquad (4.10)
\end{aligned}$$



By using the relation in the equation (4.3) and the matrix $Q$ (see (4.4)), we obtain

$$\begin{aligned}
&2(v^k - \tilde{v}^k)^T P(v^k - \bar{v}^k) \\
&\quad = \quad 2(v^k - \tilde{v}^k)^T Q(v^k - \tilde{v}^k) \\
&\quad = \quad (v^k - \tilde{v}^k)^T (Q + Q^T)(v^k - \tilde{v}^k) \\
&\quad = \quad \begin{pmatrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} 2\beta I_m & \beta I_m & -I_m \\ \beta I_m & 2\beta I_m & -I_m \\ -I_m & -I_m & \frac{2}{\beta} I_m \end{pmatrix} \begin{pmatrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \\
&\quad = \quad \beta\|(y^k - \tilde{y}^k) + (z^k - \tilde{z}^k) - \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k)\|^2 + \|v^k - \tilde{v}^k\|_D^2. \qquad (4.11)
\end{aligned}$$

Because $\tilde{y}^k = \bar{y}^k$ and $\tilde{z}^k = \bar{z}^k$, it follows from and (4.5) that

$$\beta\|(y^k - \tilde{y}^k) + (z^k - \tilde{z}^k) - \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k)\|^2 = \frac{1}{\beta}\|\lambda^k - \bar{\lambda}^k\|^2. \qquad (4.12)$$

Combining (4.10), (4.11) and (4.12) together, we obtain

$$\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2 = \alpha\Big(\frac{1}{\beta}\|\lambda^k - \bar{\lambda}^k\|^2 + \|v^k - \tilde{v}^k\|_D^2\Big) - \alpha^2\|v^k - \bar{v}^k\|_D^2. \qquad (4.13)$$

Again, because $\tilde{y}^k = \bar{y}^k$ and $\tilde{z}^k = \bar{z}^k$, we have

$$\frac{1}{\beta}\|\lambda^k - \bar{\lambda}^k\|^2 + \|v^k - \tilde{v}^k\|_D^2 = \frac{1}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2 + \|v^k - \bar{v}^k\|_D^2.$$

Substituting it in the right hand side of (4.13), we get

$$\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2 = \frac{\alpha}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2 + \alpha(1-\alpha)\|v^k - \bar{v}^k\|_D^2,$$

and thus the proof is complete. $\square$

Combining the assertions in Lemma 4.3 and Lemma 4.4, we have proved the key inequality for the proposed method, namely,

$$\big(\theta(u)-\theta(\tilde{u}^k)\big)+(w-\tilde{w}^k)^T F(\tilde{w}^k)+\frac{1}{2}(\|v-v^k\|_H^2-\|v-v^{k+1}\|_H^2) \ge 0, \ \forall w \in \mathcal{W}. \tag{4.14}$$

Note that the above inequality is true for any $\alpha \in (0,1]$. Having the key inequalities in the above lemmas, the $\mathcal{O}(1/t)$ rate of convergence (in an ergodic sense) can be obtained easily.



**Theorem 4.1** *For any integer $t > 0$, we have a $\tilde{w}_t \in \mathcal{W}$ which satisfies*

$$\big(\theta(\tilde{u}_t) - \theta(u)\big) + (\tilde{w}_t - w)^T F(w) \le \frac{1}{2(t+1)}\|v - v^0\|_H^2, \ \forall w \in \mathcal{W},$$

*where*

$$\tilde{w}_t = \frac{1}{t+1}\sum_{k=0}^{t} \tilde{w}^k.$$

**Proof**. Since $F(\cdot)$ is monotone, it follows (4.14) that

$$\big(\theta(u)-\theta(\tilde{u}^k)\big)+(w-\tilde{w}^k)^T F(w)+\frac{1}{2}(\|v-v^k\|_H^2-\|v-v^{k+1}\|_H^2) \ge 0, \ \forall w \in \mathcal{W},$$

or, equivalently,

$$\big(\theta(\tilde{u}^k)-\theta(u)\big)+(\tilde{w}^k-w)^T F(w)+\frac{1}{2}(\|v-v^{k+1}\|_H^2-\|v-v^k\|_H^2) \le 0, \ \forall w \in \mathcal{W}.$$

Summing the above inequality over $k = 0, \ldots, t$, we obtain

$$\sum_{k=0}^{t}\big(\theta(\tilde{u}^k)-\theta(u)\big)+\Big(\sum_{k=0}^{t}\tilde{w}^k-\sum_{k=0}^{t}w\Big)^T F(w)+\frac{1}{2}(\|v-v^{t+1}\|_H^2-\|v-v^0\|_H^2) \le 0.$$

Dropping the term $\|v^{t+1} - v^0\|_G^2$, we get

$$\Big(\frac{1}{t+1}\sum_{k=0}^{t}\theta(\tilde{u}^k) - \theta(u)\Big) + \Big(\frac{1}{t+1}\sum_{k=0}^{t}\tilde{w}^k - w\Big)^T F(w) \le \frac{\|v - v^0\|_H^2}{2(t+1)}, \quad \forall w \in \mathcal{W}. \tag{4.15}$$

By incorporating the notation of $\tilde{w}_t$ and using

$$\theta(\tilde{u}_t^k) \le \frac{1}{t+1}\sum_{k=0}^{t}\theta(\tilde{u}^k) \qquad \text{(due to the convexity of } \theta(u)\text{)}$$

it follows from (4.15) that

$$\big(\theta(\tilde{u}_t) - \theta(u)\big) + (\tilde{w}_t - w)^T F(w) \le \frac{\|v - v^0\|_H^2}{2(t+1)}, \quad \forall w \in \mathcal{W}.$$

Hence, the proof is complete. $\square$

For given substantial compact set $\mathcal{D} \subset \mathcal{W}$, we define

$$d = \sup\{\|v - v^0\|_H \,|\, w \in \mathcal{D}\},$$

where $v^0 = (y^0, z^0, \lambda^0)$ is the initial point. Because $\varrho_k \ge \frac{1}{2}$, it follows that $\Upsilon_t \ge \frac{t+1}{2}$.



After $t$ iterations of the proposed method, we can find a $\tilde{w} \in \mathcal{W}$ such that

$$\sup_{w \in \mathcal{D}}\big\{\theta(\tilde{u}) - \theta(u)\big) + (\tilde{w} - w)^T F(w)\big\} \le \frac{d^2}{2(t+1)}.$$

The convergence rate $\mathcal{O}(1/t)$ of the proposed method is thus proved.

# 5 Convergence rate in the non-ergodic sense

If we use (3.3) with $\alpha \in (0,1)$ to update the new iterate, it follows from (3.15) that

$$\sum_{k=0}^{\infty}\|v^k - \bar{v}^k\|_D^2 \le \frac{1}{\alpha(1-\alpha)}\|v^0 - v^*\|_H^2 \quad \forall\, v^* \in \mathcal{V}^*. \tag{5.1}$$

This section will show that the sequence $\{\|v^k - v^{k+1}\|_D^2\}$ is monotonically non-increasing, *i. e.*,

$$\|v^{k+1} - \bar{v}^{k+1}\|_D^2 \le \|v^k - \bar{v}^k\|_D^2, \quad \forall\, k \ge 0. \tag{5.2}$$

Based on (5.1) and (5.2), we drive

$$\|v^k - \bar{v}^k\|_D^2 \le \frac{1}{(k+1)\alpha(1-\alpha)}\|v^0 - v^*\|_H^2, \quad \forall v^* \in \mathcal{V}^*. \tag{5.3}$$

Since $\|v^k - \bar{v}^k\|_D^2$ is viewed as the stopping criterion, we obtain the worst-case $O(1/t)$ convergence rate in a non-ergodic sense. An important relation in the coming proof is (see (3.3))

$$P^T(v^k - v^{k+1}) = \alpha D(v^k - \bar{v}^k), \tag{5.4}$$

where the matrices $P$ and $D$ are given in (2.7). Lemma 4.2 enables us to establish an important inequality in the following lemma.

**Lemma 5.1** *Let* $\{v^k\}$ *be the sequence generated by* (3.3)*, the associated sequence* $\{\tilde{w}^k\}$ *be defined by (4.1). Then, we have*

$$(\tilde{v}^k - \tilde{v}^{k+1})^T Q\{(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\} \ge 0, \tag{5.5}$$

*where* $Q$ *is given defined in (4.4).*

**Proof**. Set $w = \tilde{w}^{k+1}$ in (4.6) and use (4.3), we obtain

$$\theta(\tilde{u}^{k+1}) - \theta(\tilde{u}^k) + (\tilde{w}^{k+1} - \tilde{w}^k)^T F(\tilde{w}^k) \ge (\tilde{v}^{k+1} - \tilde{v}^k)^T Q(v^k - \tilde{v}^k). \tag{5.6}$$



Note that (4.6) is also true for $k := k+1$ and thus we have

$$\theta(u) - \theta(\tilde{u}^{k+1}) + (w - \tilde{w}^{k+1})^T F(\tilde{w}^{k+1}) \ge (v - \tilde{v}^{k+1})^T P(v^{k+1} - \bar{v}^{k+1}),$$

for all $w \in \Omega$. Set $w = \tilde{w}^k$ in the above inequality and use (4.3), we get

$$\theta(\tilde{u}^k) - \theta(\tilde{u}^{k+1}) + (\tilde{w}^k - \tilde{w}^{k+1})^T F(\bar{w}^{k+1}) \ge (\tilde{v}^k - \tilde{v}^{k+1})^T Q(v^{k+1} - \tilde{v}^{k+1}).$$

Adding this inequality with (5.6), we get assertion (5.5) due to the monotonicity of $F$. $\square$

**Lemma 5.2** *Let the sequence* $\{\bar{w}^k\}$ *be generated by* (3.1)*, and* $\{v^k\}$ *be the sequence updated by* (3.3)*. Then, we have*

$$\begin{aligned} &\alpha(v^k - \bar{v}^k)^T D\{(v^k - \bar{v}^k) - (v^{k+1} - \bar{v}^{k+1})\} \\ &\ge \frac{1}{2}\|P(v^k - \bar{v}^k) - P(v^{k+1} - \bar{v}^{k+1})\|_{(Q^{-T}+Q^{-1})}^2, \end{aligned} \tag{5.7}$$

*where the matrices* $P$ *and* $D$ *are given in (2.7) and* $Q$ *is given defined in (4.4).*

**Proof**. Adding the term

$$\{(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\}^T Q\{(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\}$$

to the both sides of (5.5), we get

$$(v^k - v^{k+1})^T Q\{(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\} \geq \frac{1}{2}\|(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\|^2_{(Q^T+Q)}. \tag{5.8}$$

By using (5.4) and (4.3), we have

$$(v^k - v^{k+1})^T = \alpha(v^k - \bar{v}^k)^T D P^{-1},$$

and

$$\{(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\} = Q^{-1}P\{(v^k - \bar{v}^k) - (v^{k+1} - \bar{v}^{k+1})\}.$$

Substituting them in (5.8), we obtain

$$\begin{aligned} &\alpha(v^k - \bar{v}^k)^T D\{(v^k - \bar{v}^k) - (v^{k+1} - \bar{v}^{k+1})\} \\ &\geq \frac{1}{2}\|Q^{-1}P(v^k - \bar{v}^k) - Q^{-1}P(v^{k+1} - \bar{v}^{k+1})\|^2_{(Q^T+Q)}. \end{aligned} \tag{5.9}$$

Because

$$Q^{-T}(Q^T + Q)Q^{-1} = Q^{-T} + Q^{-1},$$

the both right hand sides of (5.7) and (5.9) are equal. The assertion is proved. □



Finally, we are ready to show the assertion (5.2) in the following theorem.

**Theorem 5.1** *Let* $\{\bar{v}^k\}$ (3.1)*,* $\{v^k\}$ *be the sequence generated by* (3.3)*. The sequence* $\{\|v^k - v^{k+1}\|^2_D\}$ *is monotonically non-increasing.*

**Proof**. Setting $a = (v^k - \bar{v}^k)$ and $b = (v^{k+1} - \bar{v}^{k+1})$ in the identity

$$\|a\|^2_D - \|b\|^2_D = 2a^T D(a - b) - \|a - b\|^2_D,$$

we obtain

$$\begin{aligned} &\|v^k - \bar{v}^k\|^2_D - \|v^{k+1} - \bar{v}^{k+1}\|^2_D \\ &= 2(v^k - \bar{v}^k)^T D\{(v^k - \bar{v}^k) - (v^{k+1} - \bar{v}^{k+1})\} - \|(v^k - \bar{v}^k) - (v^{k+1} - \bar{v}^{k+1})\|^2_D. \end{aligned}$$

Inserting (5.7) into the first term of the right-hand side of the last equality, we obtain

$$\begin{aligned} &\|v^k - \bar{v}^k\|^2_D - \|v^{k+1} - \bar{v}^{k+1}\|^2_D \\ &\geq \frac{1}{\alpha}\|P(v^k - \bar{v}^k) - P(v^{k+1} - \bar{v}^{k+1})\|^2_{(Q^{-T}+Q^{-1})} \\ &\quad -\|P(v^k - \bar{v}^k) - P(v^{k+1} - \bar{v}^{k+1})\|^2_{(P^{-T}DP^{-1})} \\ &\geq \|P(v^k - \bar{v}^k) - P(v^{k+1} - \bar{v}^{k+1})\|^2_{\{(Q^{-T}+Q^{-1})-(P^{-T}DP^{-1})\}}. \end{aligned}$$

Notice that (see (4.4) and (2.7))

$$(Q^{-T} + Q^{-1}) - (P^{-T} D P^{-1}) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\beta} I_m & I_m \\ 0 & I_m & \beta I_m \end{pmatrix}$$

is a positive semidefinite matrix, the assertion (5.2) follows immediately and the lemma is proved. $\Box$

With (5.1) and (5.2), we derived (5.3). The worst-case $O(1/t)$ convergence rate in a non-ergodic sense for the proposed method with $\alpha \in (0,1)$ is proved.

By using (5.4), we can also use $\|v^k - v^{k+1}\|_H \leq \epsilon$ as the stop criterion. Since

$$H = P D^{-1} P^T,$$

we have

$$\|v^k - \tilde{v}^k\|_D^2 = \frac{1}{\alpha^2} \|D^{-1} P^T (v^k - v^{k+1})\|_D^2 = \frac{1}{\alpha^2} \|v^k - v^{k+1}\|_H^2.$$



Therefore, the assertions (5.1) and (5.3) can be rewritten as

$$\sum_{k=0}^{\infty} \|v^k - v^{k+1}\|_H^2 \leq \frac{\alpha}{(1-\alpha)} \|v^0 - v^*\|_H^2 \quad \forall\, v^* \in \mathcal{V}^*, \tag{5.10}$$

and

$$\|v^k - v^{k+1}\|_H^2 \leq \frac{\alpha}{(k+1)(1-\alpha)} \|v^0 - v^*\|_H^2, \quad \forall v^* \in \mathcal{V}^*, \tag{5.11}$$

respectively.

# 6 Conclusions

Because of the attractive efficiency of the well-known alternating direction method (ADM), it is of strong desire to extend the ADMM to the linearly constrained convex programming problem with three separable operators. The convergence of the direct extension of the ADMM to the problem with 3 separable parts, however, is still open. The method proposed in this lecture (with update form (3.2)) is convergent and its variety to the direct extension of ADMM is tiny. We proved its $O(1/t)$ convergence rate in an ergodic sense. The

$O(1/t)$ non-ergodic convergence rate is also proved for the method using update form (3.3) with $\alpha \in (0,1)$.

## Appendix. Why the direct extension of ADMM performs well in practice ?

In this appendix, we try to explain why the direct extension of ADMM performs well in practice. If we use the direct extension of ADMM, then $w^{k+1} = \bar{w}^k$ and thus the relation (3.11) can be written as

$$\begin{array}{rl} & (v^{k+1} - v^*)^T P(v^k - v^{k+1}) \\ \geq & \big(\theta(u^{k+1}) - \theta(u^*)\big) + (w^{k+1} - w^*)^T F(w^{k+1}) \\ & +(\lambda^k - \lambda^{k+1})^T \{(y^k - y^{k+1}) + (z^k - z^{k+1})\}. \end{array} \tag{A.1}$$

Note that $M$ is not symmetric, but $M^T + M$ is positive definite. Again, using $\bar{w}^k = w^{k+1}$, the third part of (3.7) is

$$z^{k+1} \in \mathcal{Z}, \;\; \theta_3(z) - \theta_3(z^{k+1}) + (z - z^{k+1})^T(-\lambda^{k+1}) \geq 0, \;\; \forall\, z \in \mathcal{Z}.$$

It holds also for the previous iteration, thus we have

$$z^{k} \in \mathcal{Z}, \;\; \theta_3(z) - \theta_3(z^{k}) + (z - z^{k})^T(-\lambda^{k}) \geq 0, \;\; \forall\, z \in \mathcal{Z}.$$



Setting $z = z^k$ and $z = z^{k+1}$ in the above two sub-VIs, respectively, and then add the two resulting inequalities, we obtain

$$(\lambda^k - \lambda^{k+1})^T(z^k - z^{k+1}) \geq 0. \tag{A.2}$$

Since $(w^{k+1} - w^*)^T F(w^{k+1}) = (w^{k+1} - w^*)^T F(w^*)$ and

$$P = D + \begin{pmatrix} 0 & 0 & 0 \\ \beta I_m & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \text{(see (2.7))}$$

it follows from (A.1) and (A.2) that

$$(v^{k+1} - v^*)^T D(v^k - v^{k+1}) \geq (\lambda^k - \lambda^{k+1})^T(y^k - y^{k+1}) + \Delta_k \tag{A.3}$$

where

$$\Delta_k = \big(\theta(u^{k+1}) - \theta(u^*)\big) + (w^{k+1} - w^*)^T F(w^*) + (z^{k+1} - z^*)^T \beta(y^{k+1} - y^k). \tag{A.4}$$

Therefore, by using (A.3), we obtain

$$
\begin{aligned}
&\|v^k - v^*\|_D^2 - \|v^{k+1} - v^*\|_D^2 \\
&\quad = \quad \|(v^{k+1} - v^*) + (v^k - v^{k+1})\|_D^2 - \|v^{k+1} - v^*\|_D^2 \\
&\quad = \quad 2(v^{k+1} - v^*)^T D(v^k - v^{k+1}) + \|v^k - v^{k+1}\|_D^2 \\
&\quad \geq \quad \|v^k - v^{k+1}\|_D^2 + 2(\lambda^k - \lambda^{k+1})^T(y^k - y^{k+1}) + 2\Delta_k.
\end{aligned} \tag{A.5}
$$

Because

$$
D = \begin{pmatrix} \beta I_m & 0 & 0 \\ 0 & \beta I_m & 0 \\ 0 & 0 & \frac{1}{\beta} I_m \end{pmatrix} \quad \text{and} \quad v = \begin{pmatrix} y \\ z \\ \lambda \end{pmatrix}
$$

it follows that

$$
\begin{aligned}
&\|v^k - v^{k+1}\|_D^2 + 2(\lambda^k - \lambda^{k+1})^T(y^k - y^{k+1}) \\
&\quad = \quad \beta\|z^k - z^{k+1}\|^2 + \beta\|(y^k - y^{k+1}) + \frac{1}{\beta}(\lambda^k - \lambda^{k+1})\|^2.
\end{aligned} \tag{A.6}
$$



Substituting (A.6) and (A.3) in the right hand side of (A.5), we obtain

$$
\begin{aligned}
&\|v^k - v^*\|_D^2 - \|v^{k+1} - v^*\|_D^2 \\
&\quad = \quad \beta\|z^k - z^{k+1}\|^2 + \beta\|(y^k - y^{k+1}) + \frac{1}{\beta}(\lambda^k - \lambda^{k+1})\|^2 \\
&\quad\quad\quad +2\{\big(\theta(u^{k+1}) - \theta(u^*)\big) + (w^{k+1} - w^*)^T F(w^*)\} \\
&\quad\quad\quad +2(z^{k+1} - z^*)^T \beta(y^{k+1} - y^k).
\end{aligned} \tag{A.7}
$$

In the right hand side of (A.7), the terms

$$
\beta\|z^k - z^{k+1}\|^2 + \beta\|(y^k - y^{k+1}) + \frac{1}{\beta}(\lambda^k - \lambda^{k+1})\|^2,
$$

and

$$
2\big(\theta(u^{k+1}) - \theta(u^*)\big) + 2(w^{k+1} - w^*)^T F(w^*)
$$

are non-negative. However, we do not know whether the last term of the right hand side of (A.7), *i.e.*

$$
2(z^{k+1} - z^*)^T \beta(y^{k+1} - y^k)
$$

is non-negative. It is pity that we can not show that the right hand side of (A.7) is positive. It seems that the direct extension of ADMM performs well because the right hand side of

(A.7) is positive in practice.

If the right hand side of (A.7) is positive, then the sequence $\{\|v^k - v^*\|_D\}$ is Fejèr monotone and has the contractive property.

# References


[1] C. H. Chen, B. S. He and X. M. Yuan, Matrix completion via alternating direction method, *IMA Journal of Numerical Analysis* **32**(2012), 227-245.

[2] J. Douglas and H. H. Rachford, On the numerical solution of the heat conduction problem in 2 and 3 space variables, *Transactions of the American Mathematical Society* **82** (1956), 421–439.

[3] X.J. Cai, G.Y. Gu and B. S. He *On the $O(1/t)$ convergence rate of the projection and contraction methods for variational inequalities with Lipschitz continuous monotone operators*, Comput. Optim. Appl., 57(2014), 339-363.

[4] E. Esser, M. Möller, S. Osher G. Sapiro and J. Xin, A convex model for non-negative matrix factorization and dimensionality reduction on physical space, 2011, arXiv: 1102.0844v1 [stat.ML] 4 Feb 2011.

[5] F. Facchinei and J. S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity problems, Volume I*, Springer Series in Operations Research, Springer-Verlag, 2003.

[6] D. Gabay, Applications of the method of multipliers to variational inequalities, *Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems*, edited by M. Fortin and R. Glowinski, North Holland, Amsterdam, The Netherlands, 1983, pp. 299–331.





[7] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[8] B. S. He, Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities, *Computational Optimization and Applications* **42**(2009), 195–212.

[9] B. S. He, L. Z. Liao, D. Han, and H. Yang, A new inexact alternating directions method for monontone variational inequalities, *Mathematical Programming* **92**(2002), 103–118.

[10] B. S. He, M. Tao and X.M. Yuan, Alternating direction method with Gaussian back substitution for separable convex programming, *SIAM Journal on Optimization* **22**(2012), 313-340.

[11] B. S. He, M. H. Xu, and X. M. Yuan, Solving large-scale least squares covariance matrix problems by alternating direction methods, *SIAM Journal on Matrix Analysis and Applications* **32**(2011), 136-152.

[12] B. S. He and H. Yang, Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities, *Operations Research Letters* **23**(1998), 151–161.

[13] B. S. He and X. M. Yuan, On the $O(1/t)$ convergence rate of the alternating direction method, *SIAM J. Numerical Analysis* **50**(2012), 700-709.

[14] Z. C. Lin, M. M. Chen, L. Q. Wu, and Y. Ma, The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices, *manuscript*, 2009.

[15] A. Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optim.* **15** (2004), 229–251.

[16] P. Tseng, On accelerated proximal gradient methods for convex-concave optimization, manuscript, 2008.

# 凸优化和单调变分不等式的收缩算法

## 第十七讲: 多个分离算子凸优化
## 带回代的交替方向收缩算法

Alternating direction method with back substitution for convex optimization containing more separable operators

南京大学数学系　　何炳生
hebma@nju.edu.cn

The context of this lecture is based on the publication [7]



## 1 Introduction

In the literature, the alternating direction method (ADM) proposed originally for the following linearly constrained separable convex programming whose objective function is separable into two individual convex functions without crossed variables:

$$\begin{array}{ll} \min & \theta_1(x_1) + \theta_2(x_2) \\ & A_1x_1 + A_2x_2 = b, \\ & x_1 \in \mathcal{X}_1 \quad \text{and} \quad x_2 \in \mathcal{X}_2, \end{array} \tag{1.1}$$

where $\theta_1 : \Re^{n_1} \to \Re$ and $\theta_2 : \Re^{n_2} \to \Re$ are closed proper convex functions (not necessarily smooth); $\mathcal{X}_1 \subset \Re^{n_1}$ and $\mathcal{X}_2 \subset \Re^{n_2}$ are closed convex sets; $A_1 \in \Re^{L\times n_1}$ and $A_2 \in \Re^{L\times n_2}$ are given matrices; and $b \in \Re^L$ is a given vector. The iterative scheme of ADM for solving (1.1) is as follows:

$$\begin{cases} x_1^{k+1} = \arg\min\left\{\theta_1(x_1) + \frac{\beta}{2}\|(A_1x_1 + A_2x_2^k - b) - \frac{1}{\beta}\lambda^k\|^2 \;\middle|\; x_1 \in \mathcal{X}_1\right\}, \\ x_2^{k+1} = \arg\min\left\{\theta_2(x_2) + \frac{\beta}{2}\|(A_1x_1^{k+1} + A_2x_2 - b) - \frac{1}{\beta}\lambda^k\|^2 \;\middle|\; x_2 \in \mathcal{X}_2\right\}, \\ \lambda^{k+1} = \lambda^k - \beta(A_1x_1^{k+1} + A_2x_2^{k+1} - b), \end{cases} \tag{1.2}$$

where $\lambda^k \in \Re^L$ is the Lagrange multiplier associated with the linear constraint and $\beta > 0$ is the penalty parameter for the violation of the linear constraint.

In this paper, we consider the general case of linearly constrained separable convex programming with $m \geq 3$:

$$\begin{array}{ll} \min & \sum_{i=1}^m \theta_i(x_i) \\ & \sum_{i=1}^m A_i x_i = b; \\ & x_i \in \mathcal{X}_i, \quad i = 1, \cdots, m; \end{array} \tag{1.3}$$

where $\theta_i : \Re^{n_i} \to \Re$ $(i = 1, \ldots, m)$ are closed proper convex functions (not necessarily smooth); $\mathcal{X}_i \subset \Re^{n_i}$ $(i = 1, \ldots, m)$ are closed convex sets; $A_i \in \Re^{l \times n_i}$ $(i = 1, \ldots, m)$ are given matrices and $b \in \Re^l$ is a given vector.



Because of the efficiency of ADM for (1.1), a natural idea for solving (1.3) is to extend the ADM (1.2) from the special case (1.1) to the general case (1.3).

In fact, even for the special case of (1.3) with $m = 3$, the convergence of the extended ADM is still open.

In this paper, we provide a novel approach towards the extension of ADM for the problem (1.3). More specifically, we show that if a new iterate is generated by correcting the output of the ADM with a Gaussian back substitution procedure, then the sequence of iterates is convergent to a solution of (1.3). In this sense, we prove the convergence of the extension of ADM for (1.3). The resulting method is called the ADM with Gaussian back substitution from now on.

Alternatively, the ADM with Gaussian back substitution can be regarded as a prediction-correction type method whose predictor is generated by the ADM procedure and the correction is completed by a Gaussian back substitution procedure. We prove the convergence of the ADM with Gaussian back substitution under the analytic framework of contractive type methods

Throughout, we assume that the matrices $A_i^T A_i$ $(i = 2, \ldots, m)$ are nonsingular and the solution set of (1.3) is nonempty.

## 2 The variational inequality characterization

In this section, we derive the first-order optimality condition of (1.3) and thus characterize (1.3) by a variational inequality (VI). As we will show, the VI characterization is convenient for the convergence analysis to be conducted.

By attaching a Lagrange multiplier vector $\lambda \in \Re^l$ to the linear constraint, the Lagrange function of (1.3) is:

$$L(x_1, x_2, \ldots, x_m, \lambda) = \sum_{i=1}^{m} \theta_i(x_i) - \lambda^T (\sum_{i=1}^{m} A_i x_i - b), \tag{2.1}$$

which is defined on

$$\mathcal{W} := \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m \times \Re^l.$$

Let $\big(x_1^*, x_2^*, \ldots, x_m^*, \lambda^*\big)$ be a saddle point of the Lagrange function (2.1). Then we have

$$\begin{aligned} L_{\lambda \in \Re^l}(x_1^*, x_2^*, \cdots, x_m^*, \lambda) \quad &\le \quad L(x_1^*, x_2^*, \cdots, x_m^*, \lambda^*) \\ &\le \quad L_{x_i \in \mathcal{X}_i \ (i=1,\ldots,m)}(x_1, x_2, \ldots, x_m, \lambda^*). \end{aligned}$$

For $i \in \{1, 2, \cdots, m\}$, we denote by $\partial\theta_i(x_i)$ the subdifferential of the convex function



$\theta_i(x_i)$ and by $f_i(x_i) \in \partial\theta_i(x_i)$ a given subgradient of $\theta_i(x_i)$.

It is evident that finding a saddle point of $L(x_1, x_2, \ldots, x_m, \lambda)$ is equivalent to finding $w^* = (x_1^*, x_2^*, ..., x_m^*, \lambda^*) \in \mathcal{W}$, such that

$$\left\{ \begin{array}{l} (x_1 - x_1^*)^T \{f_1(x_1^*) - A_1^T \lambda^*\} \ge 0, \\ (x_2 - x_2^*)^T \{f_2(x_2^*) - A_2^T \lambda^*\} \ge 0, \\ \qquad \vdots \\ (x_m - x_m^*)^T \{f_m(x_m^*) - A_m^T \lambda^*\} \ge 0, \\ (\lambda - \lambda^*)^T (\sum_{i=1}^{m} A_i x_i^* - b) \ge 0, \end{array} \right. \tag{2.2}$$

for all $w = (x_1, x_2, \cdots, x_m, \lambda) \in \mathcal{W}$. More compactly, (2.2) can be written into the following VI:

$$(w - w^*)^T F(w^*) \ge 0, \quad \forall\, w \in \mathcal{W}, \tag{2.3a}$$

where

$$w = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ \lambda \end{pmatrix} \quad \text{and} \quad F(w) = \begin{pmatrix} f_1(x_1) - A_1^T\lambda \\ f_2(x_2) - A_2^T\lambda \\ \vdots \\ f_m(x_m) - A_m^T\lambda \\ \sum_{i=1}^m A_i x_i - b \end{pmatrix}. \qquad (2.3b)$$

Note that the operator $F(w)$ defined in (2.3b) is monotone due to the fact that $\theta_i$'s are all convex functions. In addition, since we have assumed that the solution set of (1.3) is not empty, the solution set of (2.3), denoted by $\mathcal{W}^*$, is also nonempty.

In addition to the notation of $w = (x_1, x_2, \cdots, x_m, \lambda)$, for any integer number $k$ we also use the following notation:

$$v = (x_2, \cdots, x_m, \lambda).$$

Moreover, we define

$$\mathcal{V}^* = \{(x_2^*, \dots, x_m^*, \lambda^*) \,|\, (x_1^*, x_2^*, \dots, x_m^*, \lambda^*) \in \mathcal{W}^*\}.$$



# 3 The ADM with Gaussian back substitution

In this section, we show the combination of the extended ADM scheme (1.2) with a Gaussian back substitution procedure, and derive the resulting ADM with Gaussian back substitution for solving (1.3). We also elucidate how to realize the Gaussian back substitution for some special cases of (1.3).

To present the Gaussian back substitution procedure, we define the matrices:

$$M = \begin{pmatrix} \beta A_2^T A_2 & 0 & \cdots & \cdots & 0 \\ \beta A_3^T A_2 & \beta A_3^T A_3 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \beta A_m^T A_2 & \beta A_m^T A_3 & \cdots & \beta A_m^T A_m & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{\beta} I_l \end{pmatrix}, \qquad (3.1)$$

and

$$H = \text{diag}\big(\beta A_2^T A_2, \beta A_3^T A_3, \dots, \beta A_m^T A_m, \frac{1}{\beta} I_l\big). \qquad (3.2)$$

Note that for any $\beta > 0$, under the assumption that all the matrices $A_i^T A_i$'s are nonsingular, the matrix $M$ defined in (3.1) is a non-singular lower-triangular block matrix and $H$ defined in (3.2) is a symmetric positive definite matrix. In addition, according to (3.1) and (3.2), we easily have:

$$H^{-1}M^T = \begin{pmatrix} I_{n_2} & (A_2^TA_2)^{-1}A_2^TA_3 & \cdots & (A_2^TA_2)^{-1}A_2^TA_m & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & (A_{m-1}^TA_{m-1})^{-1}A_{m-1}^TA_m & 0 \\ 0 & \cdots & 0 & I_{n_m} & 0 \\ 0 & \cdots & 0 & 0 & I_l \end{pmatrix}. \tag{3.3}$$

which is a upper-triangular block matrix whose diagonal components are identity matrices.



**Algorithm: The ADM with Gaussian back substitution for (1.3)**:

Let $\beta > 0$ and $\alpha \in [0.5, 1)$, With the given iterate $v^k = (x_2^k, \cdots, x_m^k, \lambda^k)$.

**Step 1. ADM step (prediction step)**. For $i = 1, \ldots, m$, obtain $\tilde{x}_i^k$ in the forward (alternating) order by solving the following $x_i$-problem:

$$\min\left\{\theta_i(x) + \tfrac{\beta}{2}\|(\textstyle\sum_{j=1}^{i-1}A_j\tilde{x}_j^k + A_ix_i + \sum_{j=i+1}^{m}A_jx_j^k - b) - \tfrac{1}{\beta}\lambda^k\|^2 \mid x_i \in \mathcal{X}_i\right\} \tag{3.4a}$$

and set

$$\tilde{\lambda}^k = \lambda^k - \beta(\textstyle\sum_{j=1}^{m}A_j\tilde{x}_j^k - b). \tag{3.4b}$$

**Step 2. Gaussian back substitution step (correction step)**. Correct the ADM output $\tilde{w}^k$ in the backward order by the following Gaussian back substitution procedure and generate the new iterate $v^{k+1}$:

$$H^{-1}M^T(v^{k+1} - v^k) = \alpha\big(\tilde{v}^k - v^k\big). \tag{3.4c}$$

where the matrices $M$ and $H$ are defined by (3.1) and (3.2), respectively.

Note that in this method, the iteration is from $v^k$ to $v^{k+1}$, the variable $x_1$ is only an intermediate variable.

Recall that the matrix $H^{-1}M^T$ defined in (3.3) is a upper-triangular block matrix. The Gaussian back substitution step (3.4c) is thus very easy to execute. In fact, as we mentioned, after the predictor is generated by the ADM scheme (3.4a) in the forward (alternating) order, the proposed Gaussian back substitution step corrects the predictor in the backward order. Since the Gaussian back substitution step is easy to perform, the computation of each iteration of the ADM with Gaussian back substitution is dominated by the ADM procedure (3.4a).

To show the main idea with clearer notation, we restrict our theoretical discussion to the case with fixed $\beta > 0$.

The main task of the Gaussian back substitution step (3.4c) can be rewritten into

$$v^{k+1} = v^k - \alpha M^{-T} H(v^k - \tilde{v}^k). \tag{3.5}$$

As we will show, $-(v^k - \tilde{v}^k)$ is a descent direction of the distance function $\frac{1}{2}\|v - v^*\|_G^2$ with $G = MH^{-1}M^T$ at the point $v = v^k$ for any $v^* \in \mathcal{V}^*$. In this sense, the proposed ADM with Gaussian back substitution can also be regarded as an ADM-based contraction method where the output of the ADM scheme (3.4a) contributes a descent direction of the distance function. Thus, the constant $\alpha$ in (3.4c) plays the role of a step size along the



descent direction $-(v^k - \tilde{v}^k)$. In fact, we can choose the step size dynamically based on some techniques in the literature (e.g. [10]), and the Gaussian back substitution procedure with the constant $\alpha$ can be modified accordingly into the following variant with a dynamical step size:

$$H^{-1}M^T(v^{k+1} - v^k) = \gamma \alpha_k^* \big(\tilde{v}^k - v^k\big), \tag{3.6}$$

where

$$\alpha_k^* = \frac{\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2}{2\|v^k - \tilde{v}^k\|_H^2}; \tag{3.7}$$

$$Q = \begin{pmatrix} \beta A_2^T A_2 & \beta A_2^T A_3 & \cdots & \beta A_2^T A_m & A_2^T \\ \beta A_3^T A_2 & \beta A_3^T A_3 & \cdots & \beta A_3^T A_m & A_3^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \beta A_m^T A_2 & \beta A_m^T A_3 & \cdots & \beta A_m^T A_m & A_m^T \\ A_2 & A_3 & \cdots & A_m & \frac{1}{\beta} I_l \end{pmatrix}; \tag{3.8}$$

and $\gamma \in (0, 2)$. Indeed, for any $\beta > 0$, the symmetric matrix $Q$ is positive semi-definite.

Then, for given $v^k$ and the $\tilde{v}^k$ obtained by the ADM procedure (3.4a), we have that

$$\|v^k - \tilde{v}^k\|_H^2 = \beta \sum_{i=2}^{m} \big\|A_i(x_i^k - \tilde{x}_i^k)\big\|^2 + \frac{1}{\beta}\big\|\lambda^k - \tilde{\lambda}^k\big\|^2,$$

and

$$\|v^k - \tilde{v}^k\|_Q^2 = \beta \left\| \sum_{i=2}^{m} A_i(x_i^k - \tilde{x}_i^k) + \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k) \right\|^2,$$

where the norm $\|v\|_H^2$ ($\|v\|_Q^2$, respectively) is defined as $v^T H v$ ($v^T Q v$, respectively). In fact, it is easy to prove that the step size $\alpha_k^*$ defined in (3.7) satisfies $\frac{1}{2} \le \alpha_k^* \le \frac{m+1}{2}$.

# 4 Convergence of the proposed method

In this section, we prove the convergence of the proposed ADM with Gaussian back substitution for solving (1.3). Our proof follows the analytic framework of contractive type methods, and it consists of the following three phases:

1.) Prove that $-M^{-T}H(v^k - \tilde{v}^k)$ is a descent direction of the function $\frac{1}{2}\|v - v^*\|_G^2$ at



the point $v = v^k$ whenever $\tilde{v}^k \neq v^k$, where $\tilde{v}^k$ is generated by the ADM scheme (3.4a), $v^* \in \mathcal{V}^*$ and $G$ is a positive definite matrix.

2.) Prove that the sequence generated by the proposed ADM with Gaussian back substitution is contractive with respect to $\mathcal{V}^*$.

3.) Derive the convergence based on the Fej$\acute{e}$r monotonicity of the sequence generated by the proposed ADM with Gaussian back substitution.

Accordingly, we divide this section into three subsections to address the tasks listed above.

## 4.1 Verification of the descent directions

We mainly show that $-(v^k - \tilde{v}^k)$ is a descent direction of the function $\frac{1}{2}\|v - v^*\|_G^2$ at the point $v = v^k$ whenever $\tilde{v}^k \neq v^k$, where $\tilde{v}^k$ is generated by the ADM scheme (3.4a), $v^* \in \mathcal{V}^*$ and $G$ is a positive definite matrix. For this purpose, we first prove two lemmas.

**Lemma 4.1** *Let $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ be generated by the ADM step* (3.4a) *from the given vector $v^k = (x_2^k, \ldots, x_m^k, \lambda^k)$. Then, we have*

$$\tilde{w}^k \in \mathcal{W}, \quad (w - \tilde{w}^k)^T \{d_2(v^k, \tilde{w}^k) - d_1(v^k, \tilde{v}^k)\} \ge 0, \quad \forall\, w \in \mathcal{W}, \qquad (4.1)$$

*where*

$$d_1(v^k,\tilde v^k) = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ \beta A_2^T A_2 & 0 & \cdots & \cdots & 0 \\ \beta A_3^T A_2 & \beta A_3^T A_3 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \beta A_m^T A_2 & \beta A_m^T A_3 & \cdots & \beta A_m^T A_m & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{\beta} I_l \end{pmatrix} \begin{pmatrix} x_2^k - \tilde x_2^k \\ x_3^k - \tilde x_3^k \\ \vdots \\ x_m^k - \tilde x_m^k \\ \lambda^k - \tilde\lambda^k \end{pmatrix}, \tag{4.2}$$

*and*

$$d_2(v^k,\tilde w^k) = F(\tilde w^k) + \beta \begin{pmatrix} A_1^T \\ A_2^T \\ \vdots \\ A_m^T \\ 0 \end{pmatrix} \big(\sum_{j=2}^m A_j(x_j^k - \tilde x_j^k)\big). \tag{4.3}$$



**Proof**. Since $\tilde x_i^k$ is the solution of (3.4a), for $i = 1,2,\ldots,m$, according to the optimality condition, we have

$$\begin{aligned} \tilde x_i^k \in \mathcal{X}_i, \quad (x_i - \tilde x_i^k)^T \Big\{ & f_i(\tilde x_i^k) - A_i^T \lambda^k \\ & + \beta A_i^T \big(\textstyle\sum_{j=1}^i A \tilde x_j^k + \sum_{j=i+1}^m A_j x_j^k - b\big)\Big\} \ge 0, \ \ \forall\, x_i \in \mathcal{X}_i. \end{aligned} \tag{4.4}$$

By using the fact (see (3.4b)) $\lambda^k = \tilde\lambda^k + \beta(\sum_{j=1}^m A_j \tilde x_j^k - b)$, thus, we have

$$A_i^T \lambda^k = A_i \tilde\lambda^k + \beta A_i (\sum_{j=1}^m A_j \tilde x_j^k - b).$$

Substituting it in (4.4), we obtain

$$\begin{aligned} \tilde x_i^k \in \mathcal{X}_i, \quad (x_i - \tilde x_i^k)^T \Big\{ & f_i(\tilde x_i^k) - A_i^T \tilde\lambda^k \\ & + \beta A_i^T \big(\textstyle\sum_{j=i+1}^m A_j (x_j^k - \tilde x_j^k)\big)\Big\} \ge 0, \ \ \forall\, x_i \in \mathcal{X}_i. \end{aligned} \tag{4.5}$$

It follows from (4.5) that $\tilde{x}^k \in \mathcal{X}$ and

$$\begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \left\{ \begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T\tilde{\lambda}^k \\ f_2(\tilde{x}_2^k) - A_2^T\tilde{\lambda}^k \\ \vdots \\ f_m(\tilde{x}_m^k) - A_m^T\tilde{\lambda}^k \end{pmatrix} + \beta \begin{pmatrix} A_1^T\left(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\right) \\ A_2^T\left(\sum_{j=3}^m A_j(x_j^k - \tilde{x}_j^k)\right) \\ \vdots \\ 0 \end{pmatrix} \right\} \geq 0, \tag{4.6}$$

for all $x \in \mathcal{X}$. Adding

$$\begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \beta \begin{pmatrix} 0 \\ A_2^T\left(\sum_{j=2}^2 A_j(x_j^k - \tilde{x}_j^k)\right) \\ \vdots \\ A_m^T\left(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\right) \end{pmatrix}$$

to the both sides of (4.6), we get $\tilde{x}^k \in \mathcal{X}$ and



$$\begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T\tilde{\lambda}^k + \beta A_1^T\left(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\right) \\ f_2(\tilde{x}_2^k) - A_2^T\tilde{\lambda}^k + \beta A_2^T\left(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\right) \\ \vdots \\ f_m(\tilde{x}_m^k) - A_m^T\tilde{\lambda}^k + \beta A_m^T\left(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\right) \end{pmatrix}$$

$$\geq \begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \begin{pmatrix} 0 \\ \beta A_2^T\left(\sum_{j=2}^2 A_j(x_j^k - \tilde{x}_j^k)\right) \\ \vdots \\ \beta A_m^T\left(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\right) \end{pmatrix}, \quad \forall\, x \in \mathcal{X}. \tag{4.7}$$

Because that $\sum_{j=1}^m A_j\tilde{x}_j^k - b = \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k)$, we have

$$(\lambda - \tilde{\lambda}^k)^T(\sum_{j=1}^m A_j\tilde{x}_j^k - b) = (\lambda - \tilde{\lambda}^k)^T\frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k).$$

Adding (4.7) and the last equality together, we get $\tilde{w}^k \in \mathcal{W}$ and

$$\begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T\tilde{\lambda}^k + \beta A_1^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ f_2(\tilde{x}_2^k) - A_2^T\tilde{\lambda}^k + \beta A_2^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ \vdots \\ f_m(\tilde{x}_m^k) - A_m^T\tilde{\lambda}^k + \beta A_m^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ \sum_{i=1}^m A_i\tilde{x}_i^k - b \end{pmatrix}$$

$$\geq \begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} 0 \\ \beta A_2^T\big(\sum_{j=2}^2 A_j(x_j^k - \tilde{x}_j^k)\big) \\ \vdots \\ \beta A_m^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}, \quad \forall\, w \in \mathcal{W}.$$

Use the notations of $d_1(v^k, \tilde{v}^k)$ and $d_2(v^k, \tilde{w}^k)$, the assertion of this lemma is proved.
□

Note that the $d_1(v^k, \tilde{v}^k)$ depends only on $v^k$ and $\tilde{v}^k$, while $d_2(v^k, \tilde{w}^k)$ is determined by both $v^k$ and $\tilde{w}^k$.



**Lemma 4.2** *Let* $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ *be generated by the ADM step (3.4a) from the given vector* $v^k = (x_2^k, \ldots, x_m^k, \lambda^k)$*. Then, we have*

$$(\tilde{w}^k - w^*)^T d_1(v^k, \tilde{v}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big), \quad \forall\, v^* \in \mathcal{V}^*, \tag{4.8}$$

*where* $d_1(v^k, \tilde{v}^k)$ *is defined in* (4.2).

**Proof**. Since $w^* \in \mathcal{W}$, it follows from (4.1) that

$$(\tilde{w}^k - w^*)^T d_1(v^k, \tilde{v}^k) \geq (\tilde{w}^k - w^*)^T d_2(v^k, \tilde{w}^k). \tag{4.9}$$

We consider the right-hand side of (4.9). By using (4.3), we get

$$\begin{aligned} &(\tilde{w}^k - w^*)^T d_2(v^k, \tilde{w}^k) \\ =\;& \big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big)^T \beta\big(\sum_{j=1}^m A_j(\tilde{x}_j^k - x_j^*)\big) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k) \end{aligned} \tag{4.10}$$

Then, we look at the right-hand side of (4.10). Since $\tilde{w}^k \in \mathcal{W}$, by using the monotonicity of $F$, we have

$$(\tilde{w}^k - w^*)^T F(\tilde{w}^k) \geq 0.$$

Because that

$$\sum_{j=1}^{m} A_j x_j^* = b \qquad \text{and} \qquad \beta(\sum_{j=1}^{m} A_j \tilde{x}_j^k - b) = \lambda^k - \tilde{\lambda}^k,$$

it follows from (4.10) that

$$(\tilde{w}^k - w^*)^T d_2(v^k, \tilde{w}^k) \quad \geq \quad (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k)\big). \tag{4.11}$$

Substituting (4.11) into (4.9), the assertion (4.8) follows immediately. $\square$

Since (see (3.1) and (4.2))

$$d_1(v^k, \tilde{v}^k) = \left( \begin{array}{c} 0 \\ M(v^k - \tilde{v}^k) \end{array} \right), \tag{4.12}$$

we have

$$(\tilde{v}^k - v^*)^T M(v^k - \tilde{v}^k) = (\tilde{w}^k - w^*)^T d_1(v^k, \tilde{v}^k)$$



and consequently from (4.8) follows that

$$(\tilde{v}^k - v^*)^T M(v^k - \tilde{v}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k)\big), \quad \forall\, v^* \in \mathcal{V}^*. \tag{4.13}$$

Now, based on the last two lemmas, we are at the stage to prove the main theorem.

**Theorem 4.1** **(Main Theorem)** *Let* $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ *be generated by the ADM step* (3.4a) *from the given vector* $v^k = (x_2^k, \ldots, x_m^k, \lambda^k)$*. Then, we have*

$$(v^k - v^*)^T M(v^k - \tilde{v}^k) \geq \frac{1}{2}\|v^k - \tilde{v}^k\|_H^2 + \frac{1}{2}\|v^k - \tilde{v}^k\|_Q^2, \quad \forall\, v^* \in \mathcal{V}^*, \tag{4.14}$$

*where* $M$*,* $H$*, and* $Q$ *are defined in* (3.1)*,* (3.2) *and* (3.8)*, respectively.*

**Proof** First, for all $v^* \in \mathcal{V}^*$, it follows from (4.13) that

$$\begin{aligned} &(v^k - v^*)^T M(v^k - \tilde{v}^k) \\ &\quad \geq \quad (v^k - \tilde{v}^k)^T M(v^k - \tilde{v}^k) + (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k)\big). \end{aligned} \tag{4.15}$$

Now, we treat the first term of (4.15). Using the matrix $M$ (see (3.1)), we have

$$
\begin{aligned}
&(v^k-\tilde v^k)^T M(v^k-\tilde v^k)\\
&= \begin{pmatrix} x_2^k-\tilde x_2^k\\ \vdots\\ x_m^k-\tilde x_m^k\\ \lambda^k-\tilde\lambda^k\end{pmatrix}
\begin{pmatrix} \beta A_2^TA_2 & 0 & \cdots & 0\\ \vdots & \ddots & \ddots & \vdots\\ \beta A_m^TA_2 & \cdots & \beta A_m^TA_m & 0\\ 0 & \cdots & 0 & \frac{1}{\beta}I_l\end{pmatrix}^T
\begin{pmatrix} x_2^k-\tilde x_2^k\\ \vdots\\ x_m^k-\tilde x_m^k\\ \lambda^k-\tilde\lambda^k\end{pmatrix}
\end{aligned}
\tag{4.16}
$$

Let us deal with the second term of the right-hand side of (4.15). By manipulations, we have

$$
\begin{aligned}
&(\lambda^k-\tilde\lambda^k)^T\big(\sum_{j=2}^m A_j(x_j^k-\tilde x_j^k)\big)\\
&= \begin{pmatrix} x_2^k-\tilde x_2^k\\ \vdots\\ x_m^k-\tilde x_m^k\\ \lambda^k-\tilde\lambda^k\end{pmatrix}^T
\begin{pmatrix} 0 & \dots & 0 & 0\\ \vdots & \ddots & \vdots & \vdots\\ 0 & \dots & 0 & 0\\ A_2 & \dots & A_m & 0\end{pmatrix}
\begin{pmatrix} x_2^k-\tilde x_2^k\\ \vdots\\ x_m^k-\tilde x_m^k\\ \lambda^k-\tilde\lambda^k\end{pmatrix}
\end{aligned}
\tag{4.17}
$$



Adding (4.16) and (4.17) together, it follows that

$$
\begin{aligned}
&(v^k-\tilde v^k)^T M(v^k-\tilde v^k)+(\lambda^k-\tilde\lambda^k)^T\big(\sum_{j=2}^m A_j(x_j^k-\tilde x_j^k)\big)\\
&= \begin{pmatrix} x_2^k-\tilde x_2^k\\ \vdots\\ x_m^k-\tilde x_m^k\\ \lambda^k-\tilde\lambda^k\end{pmatrix}
\begin{pmatrix} \beta A_2^TA_2 & 0 & \cdots & 0\\ \vdots & \ddots & \ddots & \vdots\\ \beta A_m^TA_2 & \cdots & \beta A_m^TA_m & 0\\ A_2 & \cdots & A_m & \frac{1}{\beta}I_l\end{pmatrix}^T
\begin{pmatrix} x_2^k-\tilde x_2^k\\ \vdots\\ x_m^k-\tilde x_m^k\\ \lambda^k-\tilde\lambda^k\end{pmatrix}\\
&= \frac{1}{2}\begin{pmatrix} x_2^k-\tilde x_2^k\\ \vdots\\ x_m^k-\tilde x_m^k\\ \lambda^k-\tilde\lambda^k\end{pmatrix}^T
\begin{pmatrix} 2\beta A_2^TA_2 & \cdots & \beta A_2^TA_m & A_2^T\\ \vdots & & \vdots & \vdots\\ \beta A_m^TA_2 & \cdots & 2\beta A_m^TA_m & A_m^T\\ A_2 & \cdots & A_m & \frac{2}{\beta}I_l\end{pmatrix}
\begin{pmatrix} x_2^k-\tilde x_2^k\\ \vdots\\ x_m^k-\tilde x_m^k\\ \lambda^k-\tilde\lambda^k\end{pmatrix}.
\end{aligned}
$$

Use the notation of the matrices $H$ and $Q$ to the right-hand side of the last equality, we obtain

$$
\begin{aligned}
&(v^k - \tilde{v}^k)^T M (v^k - \tilde{v}^k) + (\lambda^k - \tilde{\lambda}^k)^T \Big(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k)\Big) \\
= \quad & \frac{1}{2}\|v^k - \tilde{v}^k\|_H^2 + \frac{1}{2}\|v^k - \tilde{v}^k\|_Q^2 .
\end{aligned}
$$

Substituting the last equality in (4.15), this theorem is proved. $\Box$

It follows from (4.14) that

$$
\big\langle M H^{-1} M^T (v^k - v^*), M^{-T} H (\tilde{v}^k - v^k) \big\rangle \le -\frac{1}{2}\|v^k - \tilde{v}^k\|_{(H+Q)}^2 .
$$

In other words, by setting

$$
G = M H^{-1} M^T, \tag{4.18}
$$

$M H^{-1} M^T (v^k - v^*)$ is the gradient of the distance function $\frac{1}{2}\|v - v^*\|_G^2$, and $M^{-T} H (\tilde{v}^k - v^k)$ is a descent direction of $\frac{1}{2}\|v - v^*\|_G^2$ at the current point $v^k$ whenever $\tilde{v}^k \neq v^k$.



## 4.2 The contractive property

In this subsection, we mainly prove that the sequence generated by the proposed ADM with Gaussian back substitution is contractive with respect to the set $\mathcal{V}^*$. Note that we follow the definition of contractive type methods. With this contractive property, the convergence of the proposed ADM with Gaussian back substitution can be easily derived with subroutine analysis.

**Theorem 4.2** *Let* $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \dots, \tilde{x}_m^k, \tilde{\lambda}^k)$ *be generated by the ADM step* (3.4a) *from the given vector* $v^k = (x_2^k, \dots, x_m^k, \lambda^k)$. *Let the matrix* $G$ *be given by* (4.18). *For the new iterate* $v^{k+1}$ *produced by the Gaussian back substitution* (3.5), *there exists a constant* $c_0 > 0$ *such that*

$$
\|v^{k+1} - v^*\|_G^2 \le \|v^k - v^*\|_G^2 - c_0\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big), \quad \forall\, v^* \in \mathcal{V}^*, \tag{4.19}
$$

*where* $H$ *and* $Q$ *are defined in* (3.2) *and* (3.8), *respectively.*

**Proof** For $G = MH^{-1}M^T$ and any $\alpha \geq 0$, we obtain

$$\begin{aligned} &\|v^k - v^*\|_G^2 - \|v^{k+1} - v^*\|_G^2 \\ &= \|v^k - v^*\|_G^2 - \|(v^k - v^*) - \alpha M^{-T} H(v^k - \tilde{v}^k)\|_G^2 \\ &= 2\alpha(v^k - v^*)^T M(v^k - \tilde{v}^k) - \alpha^2 \|v^k - \tilde{v}^k\|_H^2. \end{aligned} \tag{4.20}$$

Substituting the result of Theorem 4.1 into the right-hand side of the last equation, we get

$$\begin{aligned} &\|v^k - v^*\|_G^2 - \|v^{k+1} - v^*\|_G^2 \\ &\geq \alpha\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big) - \alpha^2\|v^k - \tilde{v}^k\|_H^2 \\ &= \alpha(1-\alpha)\|v^k - \tilde{v}^k\|_H^2 + \alpha\|v^k - \tilde{v}^k\|_Q^2, \end{aligned}$$

and thus

$$\begin{aligned} &\|v^{k+1} - v^*\|_G^2 \leq \|v^k - v^*\|_G^2 \\ &\quad -\alpha\big((1-\alpha)\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big), \quad \forall\, v^* \in \mathcal{V}^*. \end{aligned} \tag{4.21}$$

Set $c_0 = \alpha(1-\alpha)$. Recall that $\alpha \in [0.5, 1)$. Thus the assertion is proved.

**Corollary 4.1** *The assertion of Theorem 4.2 also holds if the Gaussian back substitution update form is (3.6) with the calculated step length by (3.7) .*



**Proof** Analogous to the proof of Theorem 4.2, we have that

$$\begin{aligned} &\|v^k - v^*\|_G^2 - \|v^{k+1} - v^*\|_G^2 \\ &\geq 2\gamma\alpha_k^*(v^k - v^*)^T M(v^k - \tilde{v}^k) - (\gamma\alpha_k^*)^2\|v^k - \tilde{v}^k\|_H^2, \end{aligned} \tag{4.22}$$

where $\alpha_k^*$ is given by (3.7). According to (3.7), we have that

$$\alpha_k^*\big(\|v^k - \tilde{v}^k\|_H^2\big) = \frac{1}{2}\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big).$$

Then, it follows from the above equality and (4.14) that

$$\begin{aligned} &\|v^k - v^*\|_G^2 - \|v^{k+1} - v^*\|_G^2 \\ &\geq \gamma\alpha_k^*\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big) - \frac{1}{2}\gamma^2\alpha_k^*\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big) \\ &= \frac{1}{2}\gamma(2-\gamma)\alpha_k^*\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big). \end{aligned}$$

Because $\alpha_k^* \geq \frac{1}{2}$, it follows from the last inequality that

$$\begin{aligned} &\|v^{k+1} - v^*\|_G^2 \leq \|v^k - v^*\|_G^2 \\ &\quad -\frac{1}{4}\gamma(2-\gamma)\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big), \quad \forall\, v^* \in \mathcal{V}^*. \end{aligned} \tag{4.23}$$

Since $\gamma \in (0,2)$, the assertion of this corollary follows from (4.23) directly. $\square$

## 4.3 Convergence

The proposed lemmas and theorems are adequate to establish the global convergence of the proposed ADM with Gaussian back substitution, and the analytic framework is quite typical in the context of contractive type methods.

**Theorem 4.3** *Let $\{v^k\}$ and $\{\tilde{w}^k\}$ be the sequences generated by the proposed ADM with Gaussian back substitution. Then we have*

1. *The sequence $\{v^k\}$ is bounded.*
2. $\lim_{k\to\infty} \|A_i(x^k - \tilde{x}_i^k)\| = 0, i = 2,\ldots,m$, *and* $\lim_{k\to\infty} \|\lambda^k - \tilde{\lambda}^k\| = 0.$
3. *Any cluster point of $\{\tilde{w}^k\}$ is a solution point of* (2.3).
4. *The sequence $\{\tilde{v}^k\}$ converges to some $v^\infty \in \mathcal{V}^*$.*

**Proof**. The first assertion follows from (4.19) directly. From (4.19) we get

$$\textstyle\sum_{k=0}^{\infty} c_0 \|v^k - \tilde{v}^k\|_H^2 \le \|v^0 - v^*\|_G^2$$



and thus we get $\lim_{k\to\infty} \|v^k - \tilde{v}^k\|_H^2 = 0$, and consequently

$$\lim_{k\to\infty} \|A_i(x^k - \tilde{x}_i^k)\| = 0, \quad i = 2,\ldots,m, \tag{4.24}$$

and

$$\lim_{k\to\infty} \|\lambda^k - \tilde{\lambda}^k\| = 0. \tag{4.25}$$

The second assertion is proved.

Substituting (4.24) into (4.5), for $i = 1,2,\ldots,m$, we have

$$\tilde{x}_i^k \in \mathcal{X}_i, \quad \lim_{k\to\infty} (x_i - \tilde{x}_i^k)^T \{f_i(\tilde{x}_i^k) - A_i^T \tilde{\lambda}^k\} \ge 0, \quad \forall\, x_i \in \mathcal{X}_i. \tag{4.26}$$

It follows from (3.4a) and (4.25) that

$$\lim_{k\to\infty} (\sum_{j=1}^{m} A_j \tilde{x}_j^k - b) = 0. \tag{4.27}$$

Combining (4.26) and (4.27) we get

$$\tilde{w}^k \in \mathcal{W}, \quad \lim_{k\to\infty} (w - \tilde{w}^k)^T F(\tilde{w}^k) \ge 0, \quad \forall\, w \in \mathcal{W}, \tag{4.28}$$

and thus any cluster point of $\{\tilde{w}^k\}$ is a solution point of (2.3). The third assertion is proved.

It follows from the first assertion and $\lim_{k\to\infty}\|v^k - \tilde{v}^k\|_H^2 = 0$ that $\{\tilde{v}^k\}$ is also bounded. Let $v^\infty$ be a cluster point of $\{\tilde{v}^k\}$ and the subsequence $\{\tilde{v}^{k_j}\}$ converges to $v^\infty$. It follows from (4.28) that

$$\tilde{w}^{k_j} \in \mathcal{W}, \quad \lim_{k\to\infty}(w - \tilde{w}^{k_j})^T F(\tilde{w}^{k_j}) \geq 0, \quad \forall\, w \in \mathcal{W} \tag{4.29}$$

and consequently

$$\left\{\begin{array}{l} (x_i - x_i^\infty)^T\{f_i(x_i^\infty) - A_i^T\lambda^\infty\} \geq 0, \qquad \forall x_i \in \mathcal{X}_i, \;\; i = 1, \ldots, m, \\ \sum_{j=1}^m A_j x_j^\infty - b = 0. \end{array}\right. \tag{4.30}$$

This means that $v^\infty \in \mathcal{V}^*$. Since $\{v^k\}$ is $Fej\acute{e}$r monotone and $\lim_{k\to\infty}\|v^k - \tilde{v}^k\| = 0$, the sequence $\{\tilde{v}^k\}$ cannot have other cluster point and $\{\tilde{v}^k\}$ converges to $v^\infty \in \mathcal{V}^*$. $\square$

If we take $\alpha \equiv 1$ in the correction form (3.5), similarly as in the last lecture, the resulting method is convergent in the ergodic sense with the convergence rate $O(1/t)$.



# References


[1] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Nov. 2010.

[2] D. Gabay, “Applications of the Method of Multipliers to Variational Inequalities,” in *Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems*, M. Fortin and R. Glowinski, eds., North Holland, Amsterdam, The Netherlands, 299-331, 1983.

[3] D. Gabay and B. Mercier, A dual algorithm for the solution of nonlinear variational problems viafinite element approximations, Computers and Mathematics with Applications, 2:17-40, 1976.

[4] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[5] B.S. He, A class of projection and contraction methods for monotone variational inequalities, Applied Mathematics and Optimization, **35**, pp. 69-76, 1997.

[6] B.S. He, “Inexact implicit methods for monotone general variational inequalities”, Math. Program. Vol. 86, Series A, pp. 199-217, 1999

[7] B. S. He, M. Tao and X.M. Yuan, Alternating direction method with Gaussian back substitution for separable convex programming, *SIAM J. Optim. 22, 313-340, 2012*

[8] B. S. He and M.-H. Xu, A general framework of contraction methods for monotone variational inequalities, Pacific J. Optimization, 4, 195-212, 2008.

[9] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Control Optim., 14 (1976), pp. 877-898.

[10] C. H. Ye and X. M. Yuan, A Descent Method for Structured Monotone Variational Inequalities, Optimization Methods and Software, Vol. 22, pp. 329-338, 2007.

# 凸优化和单调变分不等式的收缩算法

## 第十八讲: 多个分离算子凸优化
## 带回代的线性化交替方向法

Linearized Alternating direction method with
back substitution for convex optimization
containing more separable operators

南京大学数学系　　何炳生
hebma@nju.edu.cn

The context of this lecture is based on the paper [3]



# 1 Introduction

In this paper, we consider the general case of linearly constrained separable convex programming with $m \ge 3$:

$$\begin{array}{ll} \min & \sum_{i=1}^m \theta_i(x_i) \\ & \sum_{i=1}^m A_i x_i = b; \\ & x_i \in \mathcal{X}_i, \quad i = 1, \cdots, m; \end{array} \tag{1.1}$$

where $\theta_i : \Re^{n_i} \to \Re \ (i = 1, \ldots, m)$ are closed proper convex functions (not necessarily smooth); $\mathcal{X}_i \subset \Re^{n_i} \ (i = 1, \ldots, m)$ are closed convex sets; $A_i \in \Re^{l \times n_i} \ (i = 1, \ldots, m)$ are given matrices and $b \in \Re^l$ is a given vector. Throughout, we assume that the solution set of (1.1) is nonempty.

In fact, even for the special case of (1.1) with $m = 3$, the convergence of the extended ADM is still open. In the last lecture, we provided a novel approach towards the extension of ADM for the problem (1.1). More specifically, we show that if a new iterate is generated by correcting the output of the ADM with a Gaussian back substitution procedure, then the sequence of iterates is convergent to a solution of (1.1). The resulting method is called the ADM with Gaussian back substitution (ADM-GbS).

Alternatively, the ADM-GbS can be regarded as a prediction-correction type method whose predictor is generated by the ADM procedure and the correction is completed by a Gaussian back substitution procedure. The main task of each iteration in ADM-GbS is to solve the following sub-problem:

$$\min\{\theta_i(x_i) + \tfrac{\beta}{2}\|A_i x_i - b_i\|^2 \,|\, x_i \in \mathcal{X}_i\}, \quad i = 1, \ldots, m. \tag{1.2}$$

Thus, ADM-GbS is implementable only when the subproblems of (1.2) have their solutions in the closed form. Again, each iteration of the proposed method in this lecture consists of two steps–prediction and correction. In order to implement the prediction step, we only assume that the $x_i$-subproblem

$$\min\{\theta_i(x_i) + \tfrac{r_i}{2}\|x_i - a_i\|^2 \,|\, x_i \in \mathcal{X}_i\}, \quad i = 1, \ldots, m \tag{1.3}$$

has its solution in the closed form.

The first-order optimality condition of (1.1) and thus characterize (1.1) by a variational inequality (VI). As we will show, the VI characterization is convenient for the convergence analysis to be conducted.



By attaching a Lagrange multiplier vector $\lambda \in \Re^l$ to the linear constraint, the Lagrange function of (1.1) is:

$$L(x_1, x_2, \ldots, x_m, \lambda) = \sum_{i=1}^{m} \theta_i(x_i) - \lambda^T(\sum_{i=1}^{m} A_i x_i - b), \tag{1.4}$$

which is defined on

$$\mathcal{W} := \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m \times \Re^l.$$

Let $\big(x_1^*, x_2^*, \ldots, x_m^*, \lambda^*\big)$ be a saddle point of the Lagrange function (1.4). Then we have

$$\begin{aligned} L_{\lambda\in\Re^l}(x_1^*, x_2^*, \cdots, x_m^*, \lambda) \quad &\le \quad L(x_1^*, x_2^*, \cdots, x_m^*, \lambda^*) \\ &\le \quad L_{x_i\in\mathcal{X}_i\ (i=1,\ldots,m)}(x_1, x_2, \ldots, x_m, \lambda^*). \end{aligned}$$

For $i \in \{1, 2, \cdots, m\}$, we denote by $\partial\theta_i(x_i)$ the subdifferential of the convex function $\theta_i(x_i)$ and by $f_i(x_i) \in \partial\theta_i(x_i)$ a given subgradient of $\theta_i(x_i)$.

It is evident that finding a saddle point of $L(x_1, x_2, \ldots, x_m, \lambda)$ is equivalent to finding

$w^* = (x_1^*, x_2^*, ..., x_m^*, \lambda^*) \in \mathcal{W}$, such that

$$\begin{cases} (x_1 - x_1^*)^T \{f_1(x_1^*) - A_1^T \lambda^*\} \geq 0, \\ \quad \vdots \\ (x_m - x_m^*)^T \{f_m(x_m^*) - A_m^T \lambda^*\} \geq 0, \\ (\lambda - \lambda^*)^T (\sum_{i=1}^m A_i x_i^* - b) \geq 0, \end{cases} \tag{1.5}$$

for all $w = (x_1, x_2, \cdots, x_m, \lambda) \in \mathcal{W}$. More compactly, (1.5) can be written into

$$(w - w^*)^T F(w^*) \geq 0, \quad \forall\, w \in \mathcal{W}, \tag{1.6a}$$

where

$$w = \begin{pmatrix} x_1 \\ \vdots \\ x_m \\ \lambda \end{pmatrix} \quad \text{and} \quad F(w) = \begin{pmatrix} f_1(x_1) - A_1^T \lambda \\ \vdots \\ f_m(x_m) - A_m^T \lambda \\ \sum_{i=1}^m A_i x_i - b \end{pmatrix}. \tag{1.6b}$$

Note that the operator $F(w)$ defined in (1.6b) is monotone due to the fact that $\theta_i$'s are all convex functions. In addition, the solution set of (1.6), denoted by $\mathcal{W}^*$, is also nonempty.



# 2 Linearized ADM with Gaussian back substitution

## 2.1 Linearized ADM Prediction

**Step 1. ADM step (prediction step).** Obtain $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \cdots, \tilde{x}_m^k, \tilde{\lambda}^k)$ in the forward (alternating) order by the following ADM procedure:

$$\begin{cases} \tilde{x}_1^k = \arg\min \left\{ \theta_1(x_1) + q_1^T A_1 x_1 + \frac{r_1}{2} \|x_1 - x_1^k\|^2 \;\middle|\; x_1 \in \mathcal{X}_1 \right\}; \\ \quad \vdots \\ \tilde{x}_i^k = \arg\min \left\{ \theta_i(x_i) + q_i^T A_i x_i + \frac{r_i}{2} \|x_i - x_i^k\|^2 \;\middle|\; x_i \in \mathcal{X}_i \right\}; \\ \quad \vdots \\ \tilde{x}_m^k = \arg\min \left\{ \theta_m(x_m) + q_m^T A_m x_m + \frac{r_m}{2} \|x_m - x_m^k\|^2 \;\middle|\; x_m \in \mathcal{X}_m \right\}; \\ \qquad \text{where} \quad q_i = \beta(\sum_{j=1}^{i-1} A_j \tilde{x}_j^k + \sum_{j=i}^m A_j x_j^k - b). \\ \tilde{\lambda}^k = \lambda^k - \beta(\sum_{j=1}^m A_j \tilde{x}_j^k - b). \end{cases} \tag{2.1}$$

The prediction is implementable due to the assumption (1.3) of this lecture and

$$\begin{aligned}&\arg\min\Big\{\theta_i(x_i)+ q_i^T A_i x_i + \tfrac{r_i}{2}\|x_i - x_i^k\|^2 \,\big|\, x_i\in\mathcal{X}_i\Big\}\\ &\quad = \quad \arg\min\Big\{\theta_i(x_i) + \tfrac{r_i}{2}\|x_i - \big(x_i^k - \tfrac{1}{r_i}A_i^T q_i\big)\|^2 \,\big|\, x_i\in\mathcal{X}_i\Big\}.\end{aligned}$$

> **Assumption** $r_i,\ i=1,\ldots,m$ is chosen that condition
>
> $$r_i\|x_i^k - \tilde{x}_i^k\|^2 \ge \beta\|A_i(x_i^k - \tilde{x}_i^k)\|^2 \tag{2.2}$$
>
> is satisfied in each iteration.

**In the case that** $A_i = I_{n_i}$, we take $r_i = \beta$, the condition (2.2) is satisfied. Note that in this case we have

$$\begin{aligned}&\arg\min_{x_i\in\mathcal{X}_i}\Big\{\theta_i(x_i)+ \Big\{\beta\big(\sum_{j=1}^{i-1}A_j\tilde{x}_j^k + \sum_{j=i}^{m}A_jx_j^k - b\big)\Big\}^T A_i x_i + \frac{\beta}{2}\|x_i - x_i^k\|^2\Big\}\\ &\quad = \arg\min_{x_i\in\mathcal{X}_i}\Big\{\theta_i(x_i)+ \frac{\beta}{2}\Big\|\Big(\sum_{j=1}^{i-1}A_j\tilde{x}_j^k + A_ix_i + \sum_{j=i+1}^{m}A_jx_j^k - b\Big) - \frac{1}{\beta}\lambda^k\Big\|^2\Big\}.\end{aligned}$$



## 2.2 Correction by the Gaussian back substitution

To present the Gaussian back substitution procedure, we define the matrices:

$$M = \begin{pmatrix} r_1 I_{n_1} & 0 & \cdots & \cdots & 0 \\ \beta A_2^T A_1 & r_2 I_{n_2} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \beta A_m^T A_1 & \cdots & \beta A_m^T A_{m-1} & r_m I_{n_m} & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{\beta}I_l \end{pmatrix}, \tag{2.3}$$

and

$$H = \text{diag}\big(r_1 I_{n_1}, r_2 I_{n_2}, \ldots, r_m I_{n_m}, \frac{1}{\beta}I_l\big). \tag{2.4}$$

Note that for $\beta > 0$ and $r_i > 0$, the matrix $M$ defined in (2.3) is a non-singular lower-triangular block matrix. In addition, according to (2.3) and (2.4), we have:

$$H^{-1}M^T = \begin{pmatrix} I_{n_2} & \frac{\beta}{r_1}A_1^TA_2 & \cdots & \frac{\beta}{r_1}A_1^TA_m & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & I_{n_{m-1}} & \frac{\beta}{r_{n_{m-1}}}A_{m-1}^TA_m & 0 \\ 0 & \cdots & 0 & I_{n_m} & 0 \\ 0 & \cdots & 0 & 0 & I_l \end{pmatrix}. \tag{2.5}$$

which is a upper-triangular block matrix whose diagonal components are identity matrices. The Gaussian back substitution procedure to be proposed is based on the matrix $H^{-1}M^T$ defined in (2.5).



**Step 2. Gaussian back substitution step (correction step)**. Correct the ADM output $\tilde{w}^k$ in the backward order by the following Gaussian back substitution procedure and generate the new iterate $w^{k+1}$:

$$H^{-1}M^T(w^{k+1} - w^k) = \alpha\big(\tilde{w}^k - w^k\big). \tag{2.6}$$

Recall that the matrix $H^{-1}M^T$ defined in (2.5) is a upper-triangular block matrix. The Gaussian back substitution step (2.6) is thus very easy to execute. In fact, as we mentioned, after the predictor is generated by the linearized ADM scheme (2.1) in the forward (alternating) order, the proposed Gaussian back substitution step corrects the predictor in the backward order. Since the Gaussian back substitution step is easy to perform, the computation of each iteration of the ADM with Gaussian back substitution is dominated by the ADM procedure (2.1).

To show the main idea with clearer notation, we restrict our theoretical discussion to the case with fixed $\beta > 0$. The main task of the Gaussian back substitution step (2.6) can be rewritten into

$$w^{k+1} = w^k - \alpha M^{-T}H(w^k - \tilde{w}^k). \tag{2.7}$$

As we will show, $-M^{-T}H(w^k - \tilde{w}^k)$ is a descent direction of the distance function

$\frac{1}{2}\|w - w^*\|_G^2$ with $G = MH^{-1}M^T$ at the point $w = w^k$ for any $w^* \in \mathcal{W}^*$. In this sense, the proposed linearized ADM with Gaussian back substitution can also be regarded as an ADM-based contraction method where the output of the linearized ADM scheme (2.1) contributes a descent direction of the distance function. Thus, the constant $\alpha$ in (2.6) plays the role of a step size along the descent direction $-(w^k - \tilde{w}^k)$. In fact, we can choose the step size dynamically based on some techniques in the literature (e.g. [4]), and the Gaussian back substitution procedure with the constant $\alpha$ can be modified accordingly into the following variant with a dynamical step size:

$$H^{-1}M^T(w^{k+1} - w^k) = \gamma\alpha_k^*(\tilde{w}^k - w^k), \tag{2.8}$$

where

$$\alpha_k^* = \frac{\|w^k - \tilde{w}^k\|_H^2 + \|w^k - \tilde{w}^k\|_Q^2}{2\|w^k - \tilde{w}^k\|_H^2}; \tag{2.9}$$



$$Q = \begin{pmatrix} \beta A_1^T A_1 & \beta A_1^T A_2 & \cdots & \beta A_1^T A_m & A_1^T \\ \beta A_2^T A_1 & \beta A_2^T A_2 & \cdots & \beta A_2^T A_m & A_2^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \beta A_m^T A_1 & \beta A_m^T A_2 & \cdots & \beta A_m^T A_m & A_m^T \\ A_1 & A_2 & \cdots & A_m & \frac{1}{\beta} I_l \end{pmatrix}; \tag{2.10}$$

and $\gamma \in (0, 2)$. Indeed, for any $\beta > 0$, the symmetric matrix $Q$ is positive semi-definite. Then, for given $w^k$ and the $\tilde{w}^k$ obtained by the ADM procedure (2.1), we have that

$$\|w^k - \tilde{w}^k\|_H^2 = \sum_{i=1}^m r_i \|x_i^k - \tilde{x}_i^k\|^2 + \frac{1}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2,$$

and

$$\|w^k - \tilde{w}^k\|_Q^2 = \beta \left\| \sum_{i=1}^m A_i(x_i^k - \tilde{x}_i^k) + \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k) \right\|^2,$$

where the norm $\|w\|_H^2$ ($\|w\|_Q^2$, respectively) is defined as $w^T H w$ ($w^T Q w$, respectively). Note that the step size $\alpha_k^*$ defined in (2.9) satisfies $\alpha_k^* \geq \frac{1}{2}$.

# 3 Convergence of the Linearized ADM-GbS

In this section, we prove the convergence of the proposed ADM with Gaussian back substitution for solving (1.1). Our proof follows the analytic framework of contractive type methods. Accordingly, we divide this section into three subsections.

## 3.1 Verification of the descent directions

In this subsection, we mainly show that $-(w^k - \tilde{w}^k)$ is a descent direction of the function $\frac{1}{2}\|w - w^*\|_G^2$ at the point $w = w^k$ whenever $\tilde{w}^k \neq w^k$, where $\tilde{w}^k$ is generated by the ADM scheme (2.1), $w^* \in \mathcal{W}^*$ and $G$ is a positive definite matrix.

**Lemma 3.1** *Let $\tilde{w}^k = (\tilde{x}_1^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ be generated by the linearized ADM step* (2.1) *from the given vector $w^k = (x_1^k, \ldots, x_m^k, \lambda^k)$. Then, we have*

$$\tilde{w}^k \in \mathcal{W}, \quad (w - \tilde{w}^k)^T\{d_2(w^k, \tilde{w}^k) - d_1(w^k, \tilde{w}^k)\} \geq 0, \quad \forall\, w \in \mathcal{W}, \tag{3.1}$$



*where*

$$d_1(w^k, \tilde{w}^k) = \begin{pmatrix} r_1 I_{n_1} & 0 & \cdots & \cdots & 0 \\ \beta A_2^T A_1 & r_2 I_{n_2} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \beta A_m^T A_1 & \cdots & \beta A_m^T A_{m-1} & r_m I_{n_m} & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{\beta} I_l \end{pmatrix} \begin{pmatrix} x_1^k - \tilde{x}_1^k \\ x_2^k - \tilde{x}_2^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}, \tag{3.2}$$

$$d_2(w^k, \tilde{w}^k) = F(\tilde{w}^k) + \beta \begin{pmatrix} A_1^T \\ A_2^T \\ \vdots \\ A_m^T \\ 0 \end{pmatrix} \Big(\sum_{j=1}^m A_j(x_j^k - \tilde{x}_j^k)\Big). \tag{3.3}$$

**Proof**. Since $\tilde{x}_i^k$ is the solution of (2.1), for $i = 1, 2, \ldots, m$, according to the optimality condition, we have

$$\tilde{x}_i^k \in \mathcal{X}_i, \;\; (x_i - \tilde{x}_i^k)^T \big\{ f_i(\tilde{x}_i^k) - A_i^T[\lambda^k - \beta(\textstyle\sum_{j=1}^{i-1} A\tilde{x}_j^k + \sum_{j=i}^{m} A_j x_j^k - b)] \\ + r_i(\tilde{x}_i^k - x_i^k) \big\} \geq 0, \quad \forall\, x_i \in \mathcal{X}_i. \tag{3.4}$$

By using the fact

$$\tilde{\lambda}^k = \lambda^k - \beta(\sum_{j=1}^{m} A_j \tilde{x}_j^k - b),$$

the inequality (3.4) can be written as

$$\tilde{x}_i^k \in \mathcal{X}_i, \quad (x_i - \tilde{x}_i^k)^T \big\{ f_i(\tilde{x}_i^k) - A_i^T \tilde{\lambda}^k + \beta A_i^T \big( \sum_{j=i}^{m} A_j (x_j^k - \tilde{x}_j^k) \big) \\ + r_i(\tilde{x}_i^k - x_i^k) \big\} \geq 0, \;\; \forall\, x_i \in \mathcal{X}_i. \tag{3.5}$$



Summing the inequality (3.5) over $i = 1, \ldots, m$, we obtain $\tilde{x}^k \in \mathcal{X}$ and

$$\begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \left\{ \begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T \tilde{\lambda}^k \\ f_2(\tilde{x}_2^k) - A_2^T \tilde{\lambda}^k \\ \vdots \\ f_m(\tilde{x}_m^k) - A_m^T \tilde{\lambda}^k \end{pmatrix} + \beta \begin{pmatrix} A_1^T \big( \sum_{j=1}^{m} A_j (x_j^k - \tilde{x}_j^k) \big) \\ A_2^T \big( \sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k) \big) \\ \vdots \\ A_m^T \big( A_m (x_m^k - \tilde{x}_m^k) \big) \end{pmatrix} \right\}$$

$$\geq \begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \begin{pmatrix} r_1 I_{n_1} & 0 & 0 & 0 \\ 0 & r_2 I_{n_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_m I_{n_m} \end{pmatrix} \begin{pmatrix} x_1^k - \tilde{x}_1^k \\ x_2^k - \tilde{x}_2^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \end{pmatrix} \tag{3.6}$$

for all $x \in \mathcal{X}$. Adding the following term

$$\begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \beta \begin{pmatrix} 0 \\ A_2^T \left(\sum_{j=1}^{1} A_j (x_j^k - \tilde{x}_j^k)\right) \\ \vdots \\ A_m^T \left(\sum_{j=1}^{m-1} A_j (x_j^k - \tilde{x}_j^k)\right) \end{pmatrix}$$

to the both sides of (3.6), we get $\tilde{x}^k \in \mathcal{X}$ and for all $x \in \mathcal{X}$,

$$\begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T \tilde{\lambda}^k + \beta A_1^T \left(\sum_{j=1}^{m} A_j (x_j^k - \tilde{x}_j^k)\right) \\ f_2(\tilde{x}_2^k) - A_2^T \tilde{\lambda}^k + \beta A_2^T \left(\sum_{j=1}^{m} A_j (x_j^k - \tilde{x}_j^k)\right) \\ \vdots \\ f_m(\tilde{x}_m^k) - A_m^T \tilde{\lambda}^k + \beta A_m^T \left(\sum_{j=1}^{m} A_j (x_j^k - \tilde{x}_j^k)\right) \end{pmatrix}$$

$$\geq \begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \left\{ \begin{pmatrix} r_1(x_1^k - \tilde{x}_1^k) \\ r_2(x_2^k - \tilde{x}_2^k) \\ \vdots \\ r_m(x_m^k - \tilde{x}_m^k) \end{pmatrix} + \begin{pmatrix} 0 \\ \beta A_2^T \left(\sum_{j=1}^{1} A_j (x_j^k - \tilde{x}_j^k)\right) \\ \vdots \\ \beta A_m^T \left(\sum_{j=1}^{m-1} A_j (x_j^k - \tilde{x}_j^k)\right) \end{pmatrix} \right\} \tag{3.7}$$



Because that $\sum_{j=1}^{m} A_j \tilde{x}_j^k - b = \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k)$, we have

$$(\lambda - \tilde{\lambda}^k)^T (\sum_{j=1}^{m} A_j \tilde{x}_j^k - b) = (\lambda - \tilde{\lambda}^k)^T \tfrac{1}{\beta}(\lambda^k - \tilde{\lambda}^k).$$

Adding (3.8) and the last equality together, we get $\tilde{w}^k \in \mathcal{W}$, and for all $w \in \mathcal{W}$

$$\begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T \tilde{\lambda}^k + \beta A_1^T \left(\sum_{j=1}^{m} A_j (x_j^k - \tilde{x}_j^k)\right) \\ f_2(\tilde{x}_2^k) - A_2^T \tilde{\lambda}^k + \beta A_2^T \left(\sum_{j=1}^{m} A_j (x_j^k - \tilde{x}_j^k)\right) \\ \vdots \\ f_m(\tilde{x}_m^k) - A_m^T \tilde{\lambda}^k + \beta A_m^T \left(\sum_{j=1}^{m} A_j (x_j^k - \tilde{x}_j^k)\right) \\ \sum_{j=1}^{m} A_j \tilde{x}_j^k - b \end{pmatrix}$$

$$\geq \begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_m - \tilde{x}_m^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} r_1(x_1^k - \tilde{x}_1^k) \\ r_2(x_2^k - \tilde{x}_2^k) \\ \vdots \\ r_m(x_m^k - \tilde{x}_m^k) \\ \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix} + \begin{pmatrix} 0 \\ \beta A_2^T \left(\sum_{j=1}^{1} A_j (x_j^k - \tilde{x}_j^k)\right) \\ \vdots \\ \beta A_m^T \left(\sum_{j=1}^{m-1} A_j (x_j^k - \tilde{x}_j^k)\right) \\ 0 \end{pmatrix} \right\} \tag{3.8}$$

Use the notations of $d_1(w^k, \tilde{w}^k)$ and $d_2(w^k, \tilde{w}^k)$, the assertion is proved. $\square$

**Lemma 3.2** *Let $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ be generated by the ADM step (2.1) from the given vector $w^k = (x_2^k, \ldots, x_m^k, \lambda^k)$. Then, we have*

$$(\tilde{w}^k - w^*)^T d_1(w^k, \tilde{w}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=1}^m A_j(x_j^k - \tilde{x}_j^k)\big), \quad \forall\, w^* \in \mathcal{W}^*, \tag{3.9}$$

*where $d_1(w^k, \tilde{w}^k)$ is defined in* (3.2).

**Proof**. Since $w^* \in \mathcal{W}$, it follows from (3.1) that

$$(\tilde{w}^k - w^*)^T d_1(w^k, \tilde{w}^k) \geq (\tilde{w}^k - w^*)^T d_2(w^k, \tilde{w}^k). \tag{3.10}$$

We consider the right-hand side of (3.10). By using (3.3), we get

$$\begin{aligned}&(\tilde{w}^k - w^*)^T d_2(w^k, \tilde{w}^k) \\ &= \big(\sum_{j=1}^m A_j(x_j^k - \tilde{x}_j^k)\big)^T \beta\big(\sum_{j=1}^m A_j(\tilde{x}_j^k - x_j^*)\big) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k).\end{aligned} \tag{3.11}$$

Then, we look at the right-hand side of (3.11). Since $\tilde{w}^k \in \mathcal{W}$, by using the monotonicity of $F$, we have

$$(\tilde{w}^k - w^*)^T F(\tilde{w}^k) \geq 0.$$



Because that

$$\sum_{j=1}^m A_j x_j^* = b \quad \text{and} \quad \beta(\sum_{j=1}^m A_j \tilde{x}_j^k - b) = \lambda^k - \tilde{\lambda}^k,$$

it follows from (3.11) that

$$(\tilde{w}^k - w^*)^T d_2(w^k, \tilde{w}^k) \quad \geq \quad (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big). \tag{3.12}$$

Substituting (3.12) into (3.10), the assertion (3.9) follows immediately. $\square$

Since (see (2.3) and (3.2))

$$d_1(w^k, \tilde{w}^k) = M(w^k - \tilde{w}^k), \tag{3.13}$$

from (3.9) follows that

$$(\tilde{w}^k - w^*)^T M(w^k - \tilde{w}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=1}^m A_j(x_j^k - \tilde{x}_j^k)\big), \quad \forall\, w^* \in \mathcal{W}^*. \tag{3.14}$$

Now, based on the last two lemmas, we are at the stage to prove the main theorem.

**Theorem 3.1** **(Main Theorem)** *Let* $\tilde{w}^k = (\tilde{x}_1^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ *be generated by the ADM step* (2.1) *from the given vector* $w^k = (x_1^k, \ldots, x_m^k, \lambda^k)$*. Then, we have*

$$\begin{aligned}&(w^k - w^*)^T M(w^k - \tilde{w}^k)\\ \geq\ & \frac{1}{2}\|w^k - \tilde{w}^k\|_H^2 + \frac{1}{2}\|w^k - \tilde{w}^k\|_Q^2, \ \forall\, w^* \in \mathcal{W}^*,\end{aligned} \tag{3.15}$$

*where* $M$*,* $H$*, and* $Q$ *are defined in* (2.3)*,* (2.4) *and* (2.10)*, respectively.*

**Proof** First, it follows from (3.14) that

$$\begin{aligned}&(w^k - w^*)^T M(w^k - \tilde{w}^k)\\ \geq\ & (w^k - \tilde{w}^k)^T M(w^k - \tilde{w}^k) + (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=1}^m A_j(x_j^k - \tilde{x}_j^k)\big),\end{aligned} \tag{3.16}$$

for all $w^* \in \mathcal{W}^*$.



Now, we treat the terms of the right hand side of (3.16). Using the matrix $M$ (see (2.3)), we have

$$(w^k - \tilde{w}^k)^T M(w^k - \tilde{w}^k)$$

$$= \begin{pmatrix} x_1^k - \tilde{x}_1^k \\ x_2^k - \tilde{x}_2^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} r_1 I_{n_1} & 0 & \cdots & \cdots & 0 \\ \beta A_2^T A_1 & r_2 I_{n_2} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \beta A_m^T A_1 & \cdots & \beta A_m^T A_{m-1} & r_m I_{n_m} & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{\beta} I_l \end{pmatrix} \begin{pmatrix} x_1^k - \tilde{x}_1^k \\ x_2^k - \tilde{x}_2^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \tag{3.17}$$

For the second term of the right-hand side of (3.16), by a manipulations, we obtain

$$
\begin{aligned}
&(\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=1}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\
&= \begin{pmatrix} x_1^k - \tilde{x}_1^k \\ x_2^k - \tilde{x}_2^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T
\begin{pmatrix} 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & 0 \\ A_1 & A_2 & \ldots & A_m & 0 \end{pmatrix}
\begin{pmatrix} x_1^k - \tilde{x}_1^k \\ x_2^k - \tilde{x}_2^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \quad (3.18)
\end{aligned}
$$



Adding (3.17) and (3.18) together, it follows that

$$
\begin{aligned}
&(w^k - \tilde{w}^k)^T M (w^k - \tilde{w}^k) + (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=1}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\
&= \begin{pmatrix} x_1^k - \tilde{x}_1^k \\ x_2^k - \tilde{x}_2^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}
\begin{pmatrix} r_1 I_{n_1} & 0 & \cdots & \cdots & 0 \\ \beta A_2^T A_1 & r_2 I_{n_2} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \beta A_m^T A_1 & \cdots & \beta A_m^T A_{m-1} & r_m I_{n_m} & 0 \\ A_1 & A_2 & \cdots & A_m & \frac{1}{\beta} I_l \end{pmatrix}
\begin{pmatrix} x_1^k - \tilde{x}_1^k \\ x_2^k - \tilde{x}_2^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \\
&= \frac{1}{2} \begin{pmatrix} x_1^k - \tilde{x}_1^k \\ x_2^k - \tilde{x}_2^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}
\begin{pmatrix} 2r_1 I_{n_1} & \beta A_1^T A2 & \cdots & \beta A_1^T A_m & A_1^T \\ \beta A_2^T A_1 & 2r_2 I_{n_2} & \ddots & \vdots & A_2^T \\ \vdots & \ddots & \ddots & \beta A_{m-1}^T A_m & \vdots \\ \beta A_m^T A_1 & \cdots & \beta A_m^T A_{m-1} & 2r_m I_{n_m} & A_m^T \\ A_1 & A_2 & \cdots & A_m & \frac{2}{\beta} I_l \end{pmatrix}
\begin{pmatrix} x_1^k - \tilde{x}_1^k \\ x_2^k - \tilde{x}_2^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.
\end{aligned}
$$

Use the notation of the matrices $H$, $Q$ and the condition (2.2) to the right-hand side of the last equality, we obtain

$$
\begin{aligned}
&(w^k - \tilde{w}^k)^T M (w^k - \tilde{w}^k) + (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=1}^{m} A_j (x_j^k - \tilde{x}_j^k)\big) \\
= \;& \frac{1}{2}\|w^k - \tilde{w}^k\|_H^2 + \frac{1}{2}\|w^k - \tilde{w}^k\|_Q^2 .
\end{aligned}
$$

Substituting the last equality in (3.16), the theorem is proved. $\square$

It follows from (3.15) that

$$
\big\langle MH^{-1}M^T(w^k - w^*), M^{-T}H(\tilde{w}^k - w^k)\big\rangle \le -\frac{1}{2}\|w^k - \tilde{w}^k\|_{(H+Q)}^2 .
$$

In other words, by setting

$$
G = MH^{-1}M^T, \tag{3.19}
$$

$MH^{-1}M^T(w^k - w^*)$ is the gradient of the distance function $\frac{1}{2}\|w - w^*\|_G^2$, and $M^{-T}H(\tilde{w}^k - w^k)$ is a descent direction of $\frac{1}{2}\|w - w^*\|_G^2$ at the current point $w^k$ whenever $\tilde{w}^k \neq w^k$.



## 3.2 The contractive property

In this subsection, we mainly prove that the sequence generated by the proposed ADM with Gaussian back substitution is contractive with respect to the set $\mathcal{W}^*$. Note that we follow the definition of contractive type methods. With this contractive property, the convergence of the proposed linearized ADM with Gaussian back substitution can be easily derived with subroutine analysis.

**Theorem 3.2** *Let $\tilde{w}^k = (\tilde{x}_1^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ be generated by the ADM step* (2.1) *from the given vector $w^k = (x_1^k, \ldots, x_m^k, \lambda^k)$. Let the matrix $G$ be given by* (3.19)*. For the new iterate $w^{k+1}$ produced by the Gaussian back substitution* (2.7)*, there exists a constant $c_0 > 0$ such that*

$$
\|w^{k+1} - w^*\|_G^2 \le \|w^k - w^*\|_G^2 - c_0\big(\|w^k - \tilde{w}^k\|_H^2 + \|w^k - \tilde{w}^k\|_Q^2\big), \;\forall\, w^* \in \mathcal{W}^*, \tag{3.20}
$$

*where $H$ and $Q$ are defined in* (2.4) *and* (2.10)*, respectively.*

**Proof**. For $G = MH^{-1}M^T$ and any $\alpha \geq 0$, we obtain

$$\begin{aligned}
&\|w^k - w^*\|_G^2 - \|w^{k+1} - w^*\|_G^2 \\
&= \|w^k - w^*\|_G^2 - \|(w^k - w^*) - \alpha M^{-T}H(w^k - \tilde{w}^k)\|_G^2 \\
&= 2\alpha(w^k - w^*)^T M(w^k - \tilde{w}^k) - \alpha^2\|w^k - \tilde{w}^k\|_H^2. \qquad (3.21)
\end{aligned}$$

Substituting the result of Theorem 3.1 into the right-hand side of the last equation, we get

$$\begin{aligned}
&\|w^k - w^*\|_G^2 - \|w^{k+1} - w^*\|_G^2 \\
&\geq \alpha\big(\|w^k - \tilde{w}^k\|_H^2 + \|w^k - \tilde{w}^k\|_Q^2\big) - \alpha^2\|w^k - \tilde{w}^k\|_H^2 \\
&= \alpha(1-\alpha)\|w^k - \tilde{w}^k\|_H^2 + \alpha\|w^k - \tilde{w}^k\|_Q^2,
\end{aligned}$$

and thus

$$\begin{aligned}
&\|w^{k+1} - w^*\|_G^2 \leq \|w^k - w^*\|_G^2 \\
&\quad -\alpha\big((1-\alpha)\|w^k - \tilde{w}^k\|_H^2 + \|w^k - \tilde{w}^k\|_Q^2\big), \ \forall\, w^* \in \mathcal{W}^*. \qquad (3.22)
\end{aligned}$$

Set $c_0 = \alpha(1-\alpha)$. Recall that $\alpha \in [0.5, 1)$. The assertion is proved. $\Box$

**Corollary 3.1** *The assertion of Theorem 3.2 also holds if the Gaussian back substitution is (2.8).*



**Proof**. Analogous to the proof of Theorem 3.2, we have that

$$\begin{aligned}
&\|w^k - w^*\|_G^2 - \|w^{k+1} - w^*\|_G^2 \\
&\geq 2\gamma\alpha_k^*(w^k - w^*)^T M(w^k - \tilde{w}^k) - (\gamma\alpha_k^*)^2\|w^k - \tilde{w}^k\|_H^2, \qquad (3.23)
\end{aligned}$$

where $\alpha_k^*$ is given by (2.9). According to (2.9), we have that

$$\alpha_k^*\big(\|w^k - \tilde{w}^k\|_H^2\big) = \frac{1}{2}\big(\|w^k - \tilde{w}^k\|_H^2 + \|w^k - \tilde{w}^k\|_Q^2\big).$$

Then, it follows from the above equality and (3.15) that

$$\begin{aligned}
&\|w^k - w^*\|_G^2 - \|w^{k+1} - w^*\|_G^2 \\
&\geq \gamma\alpha_k^*\big(\|w^k - \tilde{w}^k\|_H^2 + \|w^k - \tilde{w}^k\|_Q^2\big) \\
&\quad -\frac{1}{2}\gamma^2\alpha_k^*\big(\|w^k - \tilde{w}^k\|_H^2 + \|w^k - \tilde{w}^k\|_Q^2\big) \\
&= \frac{1}{2}\gamma(2-\gamma)\alpha_k^*\big(\|w^k - \tilde{w}^k\|_H^2 + \|w^k - \tilde{w}^k\|_Q^2\big).
\end{aligned}$$

Because $\alpha_k^* \geq \frac{1}{2}$, it follows from the last inequality that

$$\|w^{k+1} - w^*\|_G^2 \leq \|w^k - w^*\|_G^2 \\ -\frac{1}{4}\gamma(2-\gamma)\big(\|w^k - \tilde{w}^k\|_H^2 + \|w^k - \tilde{w}^k\|_Q^2\big), \ \forall\, w^* \in \mathcal{W}^*. \quad (3.24)$$

Since $\gamma \in (0,2)$, the assertion of this corollary follows from (3.24) directly. $\square$

## 3.3 Convergence

The proposed lemmas and theorems are adequate to establish the global convergence of the proposed ADM with Gaussian back substitution, and the analytic framework is quite typical in the context of contractive type methods.

**Theorem 3.3** *Let $\{w^k\}$ and $\{\tilde{w}^k\}$ be the sequences generated by the proposed ADM with Gaussian back substitution. Then we have*

1. *The sequence $\{w^k\}$ is bounded.*
2. *$\lim_{k\to\infty} \|w^k - \tilde{w}^k\| = 0$,*



3. *Any cluster point of $\{\tilde{w}^k\}$ is a solution point of* (1.6).
4. *The sequence $\{\tilde{w}^k\}$ converges to some $w^\infty \in \mathcal{W}^*$.*

**Proof**. The first assertion follows from (3.20) directly. In addition, from (3.20) we get

$$\sum_{k=0}^{\infty} c_0 \|w^k - \tilde{w}^k\|_H^2 \leq \|w^0 - w^*\|_G^2$$

and thus we get $\lim_{k\to\infty} \|w^k - \tilde{w}^k\|_H^2 = 0$, and consequently

$$\lim_{k\to\infty} \|x_i^k - \tilde{x}_i^k\| = 0, \quad i = 2, \ldots, m, \quad (3.25)$$

and

$$\lim_{k\to\infty} \|\lambda^k - \tilde{\lambda}^k\| = 0. \quad (3.26)$$

The second assertion is proved.

Substituting (3.25) into (3.5), for $i = 1, 2, \ldots, m$, we have

$$\tilde{x}_i^k \in \mathcal{X}_i, \quad \lim_{k\to\infty} (x_i - \tilde{x}_i^k)^T \big\{ f_i(\tilde{x}_i^k) - A_i^T \tilde{\lambda}^k \big\} \geq 0, \quad \forall\, x_i \in \mathcal{X}_i. \quad (3.27)$$

It follows from (2.1) and (3.26) that

$$\lim_{k\to\infty}(\sum_{j=1}^{m} A_j\tilde{x}_j^k - b) = 0. \tag{3.28}$$

Combining (3.27) and (3.28) we get

$$\tilde{w}^k \in \mathcal{W}, \quad \lim_{k\to\infty}(w-\tilde{w}^k)^T F(\tilde{w}^k) \ge 0, \quad \forall\, w \in \mathcal{W}, \tag{3.29}$$

and thus any cluster point of $\{\tilde{w}^k\}$ is a solution point of (1.6). The third assertion is proved.

It follows from the first assertion and $\lim_{k\to\infty}\|w^k - \tilde{w}^k\|_H^2 = 0$ that $\{\tilde{w}^k\}$ is also bounded. Let $w^\infty$ be a cluster point of $\{\tilde{w}^k\}$ and the subsequence $\{\tilde{v}^{k_j}\}$ converges to $w^\infty$. It follows from (3.29) that

$$\tilde{w}^{k_j} \in \mathcal{W}, \quad \lim_{k\to\infty}(w-\tilde{w}^{k_j})^T F(\tilde{w}^{k_j}) \ge 0, \quad \forall\, w \in \mathcal{W} \tag{3.30}$$



and consequently

$$\left\{\begin{array}{l} (x_i - x_i^\infty)^T\{f_i(x_i^\infty) - A_i^T\lambda^\infty\} \ge 0, \ \ \forall x_i \in \mathcal{X}_i, \ i = 1,\dots,m, \\ \sum_{j=1}^m A_j x_j^\infty - b = 0. \end{array}\right.$$

This means that $w^\infty \in \mathcal{W}^*$ is a solution point of (1.6) .

Since $\{w^k\}$ is Fejér monotone and $\lim_{k\to\infty}\|w^k - \tilde{w}^k\| = 0$, the sequence $\{\tilde{w}^k\}$ cannot have other cluster point and $\{\tilde{w}^k\}$ converges to $w^\infty \in \mathcal{W}^*$. $\Box$

# References


[1] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Nov. 2010.

[2] B.S. He, M. Tao and X.M. Yuan, Alternating Direction Method with Gaussian Back Substitution for Separable Convex Programming, SIAM J. Optim. 22, 313-340, 2012

[3] B.S. He and X. M. Yuan: Linearized Alternating Direction Method with Gaussian Back Substitution for Separable Convex Programming. http://www.optimization-online.org/DB_HTML/2011/10/3192.html. Numerical Algebra, Control and Optimization (Special Issue in Honor of Prof. Xuchu He's 90th Birthday), 3(2), 247-260, 2013.

[4] C. H. Ye and X. M. Yuan, A Descent Method for Structured Monotone Variational Inequalities, Optimization Methods and Software, **22**, 329-338, 2007.

# 凸优化和单调变分不等式的收缩算法

## 第十九讲: Lipschitz 连续的单调变分不等式投影收缩算法的收敛速率

Convergence rate of the PC methods
for Lipschitz continuous monotone VIs

南京大学数学系　何炳生
hebma@nju.edu.cn

The context of this lecture is based on the publication [2]



In 2005, Nemirovski's analysis indicates that the extragradient method has the $O(1/t)$ convergence rate for variational inequalities with Lipschitz continuous monotone operators. For the same problems, in the last decades, we have developed a class of Fejér monotone projection and contraction methods. Until now, only convergence results are available to these projection and contraction methods, though the numerical experiments indicate that they always outperform the extragradient method. The reason is that the former benefits from the 'optimal' step size in the contraction sense. In this paper, we prove the convergence rate under a unified conceptual framework, which includes the projection and contraction methods as special cases and thus perfects the theory of the existing projection and contraction methods. Preliminary numerical results demonstrate that the projection and contraction methods converge twice faster than the extragradient method.

## 1 Introduction

Let $\Omega$ be a nonempty closed convex subset of $\Re^n$, $F$ be a continuous mapping from $\Re^n$ to itself. The variational inequality problem, denoted by VI$(\Omega, F)$, is to find a vector

$u^* \in \Omega$ such that

$$\text{VI}(\Omega, F) \qquad (u - u^*)^T F(u^*) \geq 0, \qquad \forall u \in \Omega. \tag{1.1}$$

Notice that VI$(\Omega, F)$ is invariant when $F$ is multiplied by some positive scalar $\beta > 0$. It is well known that, for any $\beta > 0$,

$$u^* \text{ is a solution of VI}(\Omega, F) \quad \Longleftrightarrow \quad u^* = P_\Omega[u^* - \beta F(u^*)], \tag{1.2}$$

where $P_\Omega(\cdot)$ denotes the projection onto $\Omega$ with respect to the Euclidean norm, *i.e.*,

$$P_\Omega(v) = \text{argmin}\{\|u - v\| \mid u \in \Omega\}.$$

Throughout this paper we assume that the mapping $F$ is monotone and Lipschitz continuous, *i.e.*,

$$(u - v)^T (F(u) - F(v)) \geq 0, \quad \forall\, u, v \in \Re^n,$$

and there is a constant $L > 0$ (not necessary known), such that

$$\|F(u) - F(v)\| \leq L\|u - v\|, \quad \forall\, u, v \in \Re^n.$$



Moreover, we assume that the solution set of VI$(\Omega, F)$, denoted by $\Omega^*$, is nonempty. The nonempty assumption of the solution set, together with the monotonicity assumption of $F$, implies that $\Omega^*$ is closed and convex (see pp. 158 in [3]).

Among the algorithms for monotone variational inequalities, the extragradient (EG) method proposed by Korpelevich [9] is one of the attractive methods. In fact, each iteration of the extragradient method can be divided into two steps. The $k$-th iteration of EG method begins with a given $u^k \in \Omega$, the first step produces a vector $\tilde{u}^k$ via a projection

$$\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)], \tag{1.3a}$$

where $\beta_k > 0$ is selected to satisfy

$$\beta_k \|F(u^k) - F(\tilde{u}^k)\| \leq \nu \|u^k - \tilde{u}^k\|, \quad \nu \in (0, 1). \tag{1.3b}$$

Since $\tilde{u}^k$ is not accepted as the new iterate, for designation convenience, we call it as a *predictor* and $\beta_k$ is named the *prediction step size*. The second step (correction step) of the $k$-th iteration updates the new iterate $u^{k+1}$ by

$$u^{k+1} = P_\Omega[u^k - \beta_k F(\tilde{u}^k)], \tag{1.4}$$

where $\beta_k$ is called the *correction step size*. The sequence $\{u^k\}$ generated by the extragradient method is Fejér monotone with respect to the solution set, namely,

$$\|u^{k+1} - u^*\|^2 \le \|u^k - u^*\|^2 - (1-\nu^2)\|u^k - \tilde{u}^k\|^2. \tag{1.5}$$

For a proof of the above contraction property, the readers may consult [3] (see pp. 1115-1118 therein). Notice that, in the extragradient method, the step size of the prediction (1.3a) and that of the correction (1.4) are equal. Thus the two steps seem like 'symmetric'.

Because of its simple iterative forms, recently, the extragradient method has been applied to solve some large optimization problems in the area of information science, such as in machine learning [15], optical network [11] and speech recognition [12], etc. In addition, Nemirovski [10] and Tseng [16] proved the $O(1/t)$ convergence rate of the extragradient method. Both in the theoretical and practical aspects, the interest in the extragradient method becomes more active.

In the last decades, we devoted our effort to develop a class of projection and contraction (PC) methods for monotone variational inequalities [5, 6, 8, 13]. Similarly as in the extragradient method, each iteration of the PC methods consists of two steps. The prediction step of PC methods produces the predictor $\tilde{u}^k$ via (1.3) just as in the extragradient method. The PC methods exploit a pair of geminate directions [7, 8] offered



by the predictor, namely, they are

$$d(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k) - \beta_k(F(u^k) - F(\tilde{u}^k)) \quad \text{and} \quad \beta_k F(\tilde{u}^k). \tag{1.6}$$

Here, both the directions are ascent directions of the unknown distance function $\frac{1}{2}\|u - u^*\|^2$ at the point $u^k$. Based on such directions, the goal of the correction step is to generate a new iterate which is more closed to the solution set. It leads to choosing the 'optimal' step length

$$\varrho_k = \frac{(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k)}{\|d(u^k, \tilde{u}^k)\|^2}, \tag{1.7}$$

and a relaxation factor $\gamma \in (0, 2)$, the second step (*correction step*) of the PC methods updates the new iterate $u^{k+1}$ by

$$u^{k+1} = u^k - \gamma \varrho_k d(u^k, \tilde{u}^k), \tag{1.8}$$

or

$$u^{k+1} = P_\Omega[u^k - \gamma \varrho_k \beta_k F(\tilde{u}^k)]. \tag{1.9}$$

The PC methods (without line search) make one (or two) projection(s) on $\Omega$ at each iteration, and the distance of the iterates to the solution set monotonically converges to zero. According to the terminology in [1], these methods belong to the class of Fejér contraction methods. In fact, the only difference between the extragradient method and one of the PC methods is that they use different step sizes in the correction step (see (1.4) and (1.9)). According to our numerical experiments [6, 8], the PC methods always outperform the extragradient methods.

Stimulated by the complexity statement of the extragradient method, this paper shows the $O(1/t)$ convergence rate of the projection and contraction methods for monotone VIs. Recall that $\Omega^*$ can be characterized as (see (2.3.2) in pp. 159 of [3])

$$\Omega^* = \bigcap_{u\in\Omega} \big\{\tilde{u} \in \Omega : (u-\tilde{u})^T F(u) \geq 0\big\}.$$

This implies that $\tilde{u} \in \Omega$ is an approximate solution of VI$(\Omega, F)$ with the accuracy $\epsilon$ if it satisfies

$$\tilde{u} \in \Omega \quad \text{and} \quad \inf_{u\in\Omega} \big\{(u-\tilde{u})^T F(u)\big\} \geq -\epsilon.$$

In this paper, we show that, for given $\epsilon > 0$ and $\mathcal{D} \subset \Omega$, in $O(L/\epsilon)$ iterations the projection and contraction methods can find a $\tilde{u}$ such that

$$\tilde{u} \in \Omega \quad \text{and} \quad \sup_{u\in\mathcal{D}} \big\{(\tilde{u}-u)^T F(u)\big\} \leq \epsilon. \tag{1.10}$$

As a byproduct of the complexity analysis, we find why taking a suitable relaxation factor



$\gamma \in (1, 2)$ in the correction steps (1.8) and (1.9) of the PC methods can achieve the faster convergence.

The outline of this paper is as follows. Section 2 recalls some basic concepts in the projection and contraction methods. In Section 3, we investigate the geminate descent directions of the distance function. Section 4 shows the contraction property of the PC methods. In Section 5, we carry out the complexity analysis, which results in an $O(1/t)$ convergence rate and suggests using the large relaxation factor in the correction step of the PC methods. In Section 6, we present some numerical results to indicate the efficiency of the PC methods in comparison with the extragradient method. Finally, some conclusion remarks are addressed in the last section.

Throughout the paper, the following notational conventions are used. We use $u^*$ to denote a fixed but arbitrary point in the solution set $\Omega^*$. A superscript such as in $u^k$ refers to a specific vector and usually denotes an iteration index. For any real matrix $M$ and vector $v$, we denote the transpose by $M^T$ and $v^T$, respectively. The Euclidean norm will be denoted by $\|\cdot\|$.

## 2 Preliminaries

In this section, we summarize the basic concepts of the projection mapping and three fundamental inequalities for constructing the PC methods. Throughout this paper, we assume that the projection on $\Omega$ in the Euclidean-norm has a closed form and it is easy to be carried out. Since

$$P_\Omega(v) = \text{argmin}\{\frac{1}{2}\|u-v\|^2 \mid u \in \Omega\},$$

according to the optimal solution of the convex minimization problem, we have

$$(v - P_\Omega(v))^T(u - P_\Omega(v)) \le 0, \quad \forall\, v \in \Re^n, \forall\, u \in \Omega. \tag{2.1}$$

Consequently, for any $u \in \Omega$, it follows from (2.1) that

$$\begin{aligned} \|u-v\|^2 &= \|(u - P_\Omega(v)) - (v - P_\Omega(v))\|^2 \\ &= \|u - P_\Omega(v)\|^2 - 2(v - P_\Omega(v))^T(u - P_\Omega(v)) + \|v - P_\Omega(v)\|^2 \\ &\ge \|u - P_\Omega(v)\|^2 + \|v - P_\Omega(v)\|^2. \end{aligned}$$



Therefore, we have

$$\|u - P_\Omega(v)\|^2 \le \|u-v\|^2 - \|v - P_\Omega(v)\|^2, \quad \forall\, v \in \Re^n, \forall\, u \in \Omega. \tag{2.2}$$

For given $u$ and $\beta > 0$, let $\tilde{u} = P_\Omega[u - \beta F(u)]$ be given via a projection. We say that $\tilde{u}$ is a test-vector of VI$(\Omega, F)$ because

$$u = \tilde{u} \quad \Leftrightarrow \quad u \in \Omega^*.$$

Since $\tilde{u} \in \Omega$, it follows from (1.1) that

$$(\textbf{FI-1}) \qquad (\tilde{u} - u^*)^T \beta F(u^*) \ge 0, \quad \forall\, u^* \in \Omega^*. \tag{2.3}$$

Setting $v = u - \beta F(u)$ and $u = u^*$ in the inequality (2.1), we obtain

$$(\textbf{FI-2}) \qquad (\tilde{u} - u^*)^T \big((u - \tilde{u}) - \beta F(u)\big) \ge 0, \quad \forall\, u^* \in \Omega^*. \tag{2.4}$$

Under the assumption that $F$ is monotone we have

$$(\textbf{FI-3}) \qquad (\tilde{u} - u^*)^T \beta \big(F(\tilde{u}) - F(u^*)\big) \ge 0, \quad \forall\, u^* \in \Omega^*. \tag{2.5}$$

The inequalities (2.3), (2.4) and (2.5) play an important role in the projection and contraction methods. They were emphasized in [5] as *three fundamental inequalities* in the projection and contraction methods.

# 3 Predictor and the ascent directions

For given $u^k$, the predictor $\tilde{u}^k$ in the projection and contraction methods [5, 6, 8, 13] is produced by (1.3). Because the mapping $F$ is Lipschitz continuous (even if the constant $L > 0$ is unknown), without loss of generality, we can assume that $\inf_{k\geq 0}\{\beta_k\} \geq \beta_L > 0$ and $\beta_L = O(1/L)$. In practical computation, we can make an initial guesses of $\beta = \nu/L$ and decrease $\beta$ by a constant factor and repeat the procedure whenever (1.3b) is violated.

For any but fixed $u^* \in \Omega^*$, $(u - u^*)$ is the gradient of the unknown distance function $\frac{1}{2}\|u - u^*\|^2$ in the Euclidean-norm[a] at the point $u$. A direction $d$ is called an ascent direction of $\frac{1}{2}\|u - u^*\|^2$ at $u$ if and only if the inner-product $(u - u^*)^T d > 0$.

[a]For convenience, we only consider the distance function in the Euclidean-norm. All the results in this paper are easy to extended to the contraction of the distance function in $G$-norm where $G$ is a positive definite matrix.



## 3.1 Ascent directions by adding the fundamental inequalities

Setting $u = u^k$, $\tilde{u} = \tilde{u}^k$ and $\beta = \beta_k$ in the fundamental inequalities (2.3), (2.4) and (2.5), and adding them, we get

$$(\tilde{u}^k - u^*)^T d(u^k, \tilde{u}^k) \geq 0, \quad \forall\, u^* \in \Omega^*, \tag{3.1}$$

where

$$d(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k) - \beta_k\big(F(u^k) - F(\tilde{u}^k)\big), \tag{3.2}$$

which is the same $d(u^k, \tilde{u}^k)$ defined in (1.6). It follows from (3.1) that

$$(u^k - u^*)^T d(u^k, \tilde{u}^k) \geq (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k). \tag{3.3}$$

Note that, under the condition (1.3b), we have

$$\begin{aligned} & 2(u^k - \tilde{u}^k) d(u^k, \tilde{u}^k) - \|d(u^k, \tilde{u}^k)\|^2 \\ = \;& d(u^k, \tilde{u}^k)^T \{2(u^k - \tilde{u}^k) - d(u^k, \tilde{u}^k)\} \\ = \;& \|u^k - \tilde{u}^k\|^2 - \beta_k^2 \|F(u^k) - F(\tilde{u}^k)\|^2 \\ \geq \;& (1 - \nu^2)\|u^k - \tilde{u}^k\|^2. \end{aligned} \tag{3.4}$$

Consequently, from (3.3) and (3.4) we have

$$(u^k - u^*)^T d(u^k, \tilde{u}^k) \geq \frac{1}{2}\big(\|d(u^k, \tilde{u}^k)\|^2 + (1-\nu^2)\|u^k - \tilde{u}^k\|^2\big).$$

This means that $d(u^k, \tilde{u}^k)$ is an ascent direction of the unknown distance function $\frac{1}{2}\|u - u^*\|^2$ at the point $u^k$.

## 3.2 Geminate ascent directions

To the direction $d(u^k, \tilde{u}^k)$ defined in (3.2), there is a correlative ascent direction $\beta_k F(\tilde{u}^k)$. Use the notation of $d(u^k, \tilde{u}^k)$, the projection equation (1.3a) can be written as

$$\tilde{u}^k = P_\Omega\{\tilde{u}^k - [\beta_k F(\tilde{u}^k) - d(u^k, \tilde{u}^k)]\}. \tag{3.5a}$$

It follows that $\tilde{u}^k$ is a solution of VI$(\Omega, F)$ if and only if $d(u^k, \tilde{u}^k) = 0$. Assume that there is a constant $c > 0$ such that

$$\varrho_k = \frac{(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k)}{\|d(u^k, \tilde{u}^k)\|^2} \geq c, \quad \forall k \geq 0. \tag{3.5b}$$



In this paper, we call (3.5) with $c > 0$ *the general conditions* and the forthcoming analysis is based of these conditions. For given $u^k$, there are different ways to construct $\tilde{u}^k$ and $d(u^k, \tilde{u}^k)$ which satisfy the conditions (3.5) (see [8] for an example). If $\beta_k$ satisfies (1.3b) and $d(u^k, \tilde{u}^k)$ is given by (3.2), the general conditions (3.5) are satisfied with $c \geq \frac{1}{2}$ (see (3.4)). Note that an equivalent expression of (3.5a) is

$$\tilde{u}^k \in \Omega, \quad (u - \tilde{u}^k)^T\{\beta_k F(\tilde{u}^k) - d(u^k, \tilde{u}^k)\} \geq 0, \quad \forall\, u \in \Omega, \tag{3.6a}$$

and from (3.5b) we have

$$(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k) = \varrho_k \|d(u^k, \tilde{u}^k)\|^2. \tag{3.6b}$$

In fact, $d(u^k, \tilde{u}^k)$ and $\beta_k F(\tilde{u}^k)$ in (3.5a) are a pair of geminate directions and usually denoted by $d_1(u^k, \tilde{u}^k)$ and $d_2(u^k, \tilde{u}^k)$, respectively. In this paper, we restrict $d_2(u^k, \tilde{u}^k)$ to be $F(\tilde{u}^k)$ times a positive scalar $\beta_k$. If $d(u^k, \tilde{u}^k) = u^k - \tilde{u}^k$, then $\tilde{u}^k$ in (3.6a) is the solution of the subproblem in the $k$-th iteration when PPA applied to solve VI$(\Omega, F)$. Hence, the projection and contraction methods considered in this paper belong to the *prox-like contraction methods*.

The following lemmas tell us that both the direction $d(u^k, \tilde{u}^k)$ (for $u^k \in \Re^n$) and $F(\tilde{u}^k)$

(for $u^k \in \Omega$) are ascent directions of the function $\frac{1}{2}\|u - u^*\|^2$ whenever $u^k$ is not a solution point. The proof is similar to those in [7], for completeness sake of this paper, we restate the short proofs.

**Lemma 3.1** *Let the general conditions* (3.5) *be satisfied. Then we have*

$$(u^k - u^*)^T d(u^k, \tilde{u}^k) \geq \varrho_k \|d(u^k, \tilde{u}^k)\|^2, \quad \forall\, u^k \in \Re^n,\; u^* \in \Omega^*. \tag{3.7}$$

**Proof**. Note that $u^* \in \Omega$. By setting $u = u^*$ in (3.6a) (the equivalent expression of (3.5a)), we get

$$(\tilde{u}^k - u^*)^T d(u^k, \tilde{u}^k) \geq (\tilde{u}^k - u^*)^T \beta_k F(\tilde{u}^k) \geq 0, \;\; \forall u^* \in \Omega^*.$$

The last inequality follows from the monotonicity of $F$ and $(\tilde{u}^k - u^*)^T F(u^*) \geq 0$. Therefore,

$$(u^k - u^*)^T d(u^k, \tilde{u}^k) \geq (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k), \;\; \forall u^* \in \Omega^*.$$

The assertion (3.7) is followed from the above inequality and (3.6b) directly. $\square$



**Lemma 3.2** *Let the general conditions* (3.5) *be satisfied. If* $u^k \in \Omega$*, then we have*

$$(u^k - u^*)^T \beta_k F(\tilde{u}^k) \geq \varrho_k \|d(u^k, \tilde{u}^k)\|^2, \quad \forall\, u^* \in \Omega^*. \tag{3.8}$$

**Proof**. Since $(\tilde{u}^k - u^*)^T \beta_k F(\tilde{u}^k) \geq 0$, we have

$$(u^k - u^*)^T \beta_k F(\tilde{u}^k) \geq (u^k - \tilde{u}^k)^T \beta_k F(\tilde{u}^k), \;\; \forall\, u^* \in \Omega^*.$$

Note that because $u^k \in \Omega$, by setting $u = u^k$ in (3.6a), we get

$$(u^k - \tilde{u}^k)^T \beta_k F(\tilde{u}^k) \geq (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k).$$

From the above two inequalities follows that

$$(u^k - u^*)^T \beta_k F(\tilde{u}^k) \geq (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k), \;\; \forall u^* \in \Omega^*.$$

The assertion (3.8) is followed from the above inequality and (3.6b) directly. $\square$

Note that (3.7) holds for $u^k \in \Re^n$ while (3.8) is hold only for $u^k \in \Omega$.

# 4 Corrector in the contraction sense

Based on the pair of geminate ascent directions in (3.5), namely, $d(u^k, \tilde{u}^k)$ and $\beta_k F(\tilde{u}^k)$, we use the one of the following corrector forms to update the new iterate $u^{k+1}$:

$$\text{(Correction of PC Method-I)} \qquad u_I^{k+1} = u^k - \gamma \varrho_k d(u^k, \tilde{u}^k), \tag{4.1a}$$

or

$$\text{(Correction of PC Method-II)} \qquad u_{II}^{k+1} = P_\Omega[u^k - \gamma \varrho_k \beta_k F(\tilde{u}^k)], \tag{4.1b}$$

where $\gamma \in (0, 2)$ and $\varrho_k$ is defined in (3.5b). Note that the same step size length is used in (4.1a) and (4.1b) even if the search directions are different. Recall that $\tilde{u}^k$ is obtained via a projection, by using the correction form (4.1b), we have to make an additional projection on $\Omega$ in the PC methods. Replacing $\gamma\varrho_k$ in (4.1b) by 1, it reduces to the update form of the extragradient method (see (1.4)).

For any solution point $u^* \in \Omega^*$, we define

$$\vartheta_I(\gamma) = \|u^k - u^*\|^2 - \|u_I^{k+1} - u^*\|^2 \tag{4.2a}$$



and

$$\vartheta_{II}(\gamma) = \|u^k - u^*\|^2 - \|u_{II}^{k+1} - u^*\|^2, \tag{4.2b}$$

which measure the profit in the $k$-th iteration. The following theorem gives a lower bound of the profit function, the similar results were established in [6, 7, 8].

**Theorem 4.1** *For given $u^k$, let the general conditions* (3.5) *be satisfied. If the corrector is updated by* (4.1a) *or* (4.1b)*, then for any $u^* \in \Omega^*$ and $\gamma > 0$, we have*

$$\vartheta_I(\gamma) \geq q(\gamma), \tag{4.3}$$

*and*

$$\vartheta_{II}(\gamma) \geq q(\gamma) + \|u_I^{k+1} - u_{II}^{k+1}\|^2, \tag{4.4}$$

*respectively, where*

$$q(\gamma) = \gamma(2 - \gamma)\varrho_k^2 \|d(u^k, \tilde{u}^k)\|^2. \tag{4.5}$$

**Proof**. Using the definition of $\vartheta_I(\gamma)$ and $u_I^{k+1}$ (see (4.1a)), we have

$$\begin{aligned} \vartheta_I(\gamma) &= \|u^k - u^*\|^2 - \|u^k - u^* - \gamma\varrho_k d(u^k, \tilde{u}^k)\|^2 \\ &= 2\gamma\varrho_k (u^k - u^*)^T d(u^k, \tilde{u}^k) - \gamma^2 \varrho_k^2 \|d(u^k, \tilde{u}^k)\|^2. \end{aligned} \tag{4.6}$$

Recalling (3.7), we obtain

$$2\gamma\varrho_k(u^k - u^*)^T d(u^k, \tilde{u}^k) \ge 2\gamma\varrho_k^2\|d(u^k, \tilde{u}^k)\|^2.$$

Substituting it in (4.6) and using the definition of $q(\gamma)$, we get $\vartheta_I(\gamma) \ge q(\gamma)$ and the first assertion is proved. Now, we turn to show the second assertion. Because

$$u_{II}^{k+1} = P_\Omega[u^k - \gamma\varrho_k\beta_k F(\tilde{u}^k)],$$

and $u^* \in \Omega$, by setting $u = u^*$ and $v = u^k - \gamma\varrho_k\beta_k F(\tilde{u}^k)$ in (2.2), we have

$$\begin{aligned} \|u^* - u_{II}^{k+1}\|^2 \quad &\le \quad \|u^* - (u^k - \gamma\varrho_k\beta_k F(\tilde{u}^k))\|^2 \\ &\quad\; -\|u^k - \gamma\varrho_k\beta_k F(\tilde{u}^k) - u_{II}^{k+1}\|^2. \end{aligned} \tag{4.7}$$

Thus,

$$\begin{aligned} \vartheta_{II}(\gamma) \quad &= \quad \|u^k - u^*\|^2 - \|u_{II}^{k+1} - u^*\|^2 \\ &\ge \quad \|u^k - u^*\|^2 - \|(u^k - u^*) - \gamma\varrho_k\beta_k F(\tilde{u}^k)\|^2 \\ &\quad\quad +\|(u^k - u_{II}^{k+1}) - \gamma\varrho_k\beta_k F(\tilde{u}^k)\|^2 \\ &= \quad \|u^k - u_{II}^{k+1}\|^2 + 2\gamma\varrho_k\beta_k(u_{II}^{k+1} - u^*)^T F(\tilde{u}^k) \\ &\ge \quad \|u^k - u_{II}^{k+1}\|^2 + 2\gamma\varrho_k\beta_k(u_{II}^{k+1} - \tilde{u}^k)^T F(\tilde{u}^k). \end{aligned} \tag{4.8}$$



The last inequality in (4.8) follows from $(\tilde{u}^k - u^*)^T F(\tilde{u}^k) \ge 0$. Since $u_{II}^{k+1} \in \Omega$, by setting $u = u_{II}^{k+1}$ in (3.6a), we get

$$(u_{II}^{k+1} - \tilde{u}^k)^T\{\beta_k F(\tilde{u}^k) - d(u^k, \tilde{u}^k)\} \ge 0,$$

and consequently, substituting it in the right hand side of (4.8), we obtain

$$\begin{aligned} \vartheta_{II}(\gamma) \quad &\ge \quad \|u^k - u_{II}^{k+1}\|^2 + 2\gamma\varrho_k(u_{II}^{k+1} - \tilde{u}^k)^T d(u^k, \tilde{u}^k) \\ &= \quad \|u^k - u_{II}^{k+1}\|^2 + 2\gamma\varrho_k(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k) \\ &\quad\quad -2\gamma\varrho_k(u^k - u_{II}^{k+1})^T d(u^k, \tilde{u}^k). \end{aligned} \tag{4.9}$$

To the two crossed term in the right hand side of (4.9), we have (by using (3.6b))

$$2\gamma\varrho_k(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k) = 2\gamma\varrho_k^2\|d(u^k, \tilde{u}^k)\|^2,$$

and

$$\begin{aligned} &-2\gamma\varrho_k(u^k - u_{II}^{k+1})^T d(u^k, \tilde{u}^k) \\ &\quad = \quad \|(u^k - u_{II}^{k+1}) - \gamma\varrho_k d(u^k, \tilde{u}^k)\|^2 \\ &\quad\quad\quad -\|u^k - u_{II}^{k+1}\|^2 - \gamma^2\varrho_k^2\|d(u^k, \tilde{u}^k)\|^2, \end{aligned}$$

respectively. Substituting them in the right hand side of (4.9) and using

$$u^k - \gamma\varrho_k d(u^k, \tilde{u}^k) = u_I^{k+1},$$

we obtain

$$\begin{aligned} \vartheta_{II}(\gamma) &\geq \gamma(2-\gamma)\varrho_k^2\|d(u^k,\tilde{u}^k)\|^2 + \|u_I^{k+1} - u_{II}^{k+1}\|^2 \\ &= q(\gamma) + \|u_I^{k+1} - u_{II}^{k+1}\|^2, \end{aligned} \tag{4.10}$$

and the proof is complete. □

Note that $q(\gamma)$ is a quadratic function of $\gamma$, it reaches its maximum at $\gamma^* = 1$. In practice, $\varrho_k$ is the 'optimal' step size in (4.1) and $\gamma$ is a relaxation factor. Because $q(\gamma)$ is a lower bound of $\vartheta_I(\gamma)$ (resp. $\vartheta_{II}(\gamma)$), the desirable new iterate is updated by (4.1) with $\gamma \in [1, 2)$.

From Theorem 4.1 we obtain

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \gamma(2-\gamma)\varrho_k^2\|d(u^k,\tilde{u}^k)\|^2. \tag{4.11}$$

Convergence result follows from (4.11) directly. Due to the property (4.11) we call the methods which use different update forms in (4.1) PC Method-I and PC Method II,



respectively. Note that the assertion (4.11) is derived from the general conditions (3.5). For the PC methods using correction form (1.8) or (1.9), because $\varrho_k > \frac{1}{2}$, by using (3.6b) and (1.3b), it follows from (4.11) that

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \frac{1}{2}\gamma(2-\gamma)(1-\nu)\|u^k - \tilde{u}^k\|^2. \tag{4.12}$$

# 5 Convergence rate of the PC methods

This section proves the convergence rate of the projection and contraction methods. Recall that the base of the complexity proof is (see (2.3.2) in pp. 159 of [3])

$$\Omega^* = \bigcap_{u\in\Omega} \big\{\tilde{u} \in \Omega : (u - \tilde{u})^T F(u) \geq 0\big\}. \tag{5.1}$$

In the sequel, for given $\epsilon > 0$ and $\mathcal{D} \subset \Omega$, we focus our attention to find a $\tilde{u}$ such that

$$\tilde{u} \in \Omega \quad \text{and} \quad \sup_{u\in\mathcal{D}} (\tilde{u} - u)^T F(u) \leq \epsilon. \tag{5.2}$$

Although the PC Method I uses the update form (4.1a) and it does not guarantee that

$\{u^k\}$ belongs to $\Omega$, the sequence $\{\tilde{u}^k\} \subset \Omega$ in the PC methods with different corrector forms. Now, we prove the key inequality of the PC Method I for the complexity analysis.

**Lemma 5.1** *For given $u^k \in \Re^n$, let the general conditions* (3.5) *be satisfied. If the new iterate $u^{k+1}$ is updated by* (4.1a) *with any $\gamma > 0$, then we have*

$$(u - \tilde{u}^k)^T \gamma \varrho_k \beta_k F(\tilde{u}^k) + \frac{1}{2}\big(\|u - u^k\|^2 - \|u - u^{k+1}\|^2\big) \ge \frac{1}{2} q(\gamma), \quad \forall\, u \in \Omega, \tag{5.3}$$

*where $q(\gamma)$ is defined in* (4.5).

**Proof**. Because (due to (3.6a))

$$(u - \tilde{u}^k)^T \beta_k F(\tilde{u}^k) \ge (u - \tilde{u}^k)^T d(u^k, \tilde{u}^k), \quad \forall\, u \in \Omega,$$

and (see (4.1a))

$$\gamma \varrho_k d(u^k, \tilde{u}^k) = u^k - u^{k+1},$$

we need only to show that

$$(u - \tilde{u}^k)^T (u^k - u^{k+1}) + \frac{1}{2}\big(\|u - u^k\|^2 - \|u - u^{k+1}\|^2\big) \ge \frac{1}{2} q(\gamma), \quad \forall\, u \in \Omega. \tag{5.4}$$

To the crossed term in the left hand side of (5.4), namely $(u - \tilde{u}^k)^T (u^k - u^{k+1})$, using



an identity

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

we obtain

$$\begin{aligned} (u - \tilde{u}^k)^T (u^k - u^{k+1}) \quad = \quad & \frac{1}{2}\big(\|u - u^{k+1}\|^2 - \|u - u^k\|^2\big) \\ & + \frac{1}{2}\big(\|u^k - \tilde{u}^k\|^2 - \|u^{k+1} - \tilde{u}^k\|^2\big). \end{aligned} \tag{5.5}$$

By using $u^{k+1} = u^k - \gamma \varrho_k d(u^k, \tilde{u}^k)$ and (3.6b), we get

$$\begin{aligned} & \|u^k - \tilde{u}^k\|^2 - \|u^{k+1} - \tilde{u}^k\|^2 \\ & \quad = \quad \|u^k - \tilde{u}^k\|^2 - \|(u^k - \tilde{u}^k) - \gamma \varrho_k d(u^k, \tilde{u}^k)\|^2 \\ & \quad = \quad 2\gamma \varrho_k (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k) - \gamma^2 \varrho_k^2 \|d(u^k, \tilde{u}^k)\|^2 \\ & \quad = \quad \gamma(2 - \gamma) \varrho_k^2 \|d(u^k, \tilde{u}^k)\|^2. \end{aligned}$$

Substituting it in the right hand side of (5.5) and using the definition of $q(\gamma)$, we obtain (5.4) and the lemma is proved. $\square$

The both sequences $\{\tilde{u}^k\}$ and $\{u^k\}$ in the PC method II belong to $\Omega$. In the following lemma we prove the same assertion for PC method II as in Lemma 5.1.

**Lemma 5.2** *For given $u^k \in \Omega$, let the general conditions* (3.5) *be satisfied. If the new iterate $u^{k+1}$ is updated by* (4.1b) *with any $\gamma > 0$, then we have*

$$(u - \tilde{u}^k)^T \gamma \varrho_k \beta_k F(\tilde{u}^k) + \frac{1}{2}\big(\|u - u^k\|^2 - \|u - u^{k+1}\|^2\big) \geq \frac{1}{2} q(\gamma), \ \ \forall\, u \in \Omega, \tag{5.6}$$

*where $q(\gamma)$ is defined in* (4.5).

**Proof**. For investigating $(u - \tilde{u}^k)^T \beta_k F(\tilde{u}^k)$, we divide it in the terms

$$(u^{k+1} - \tilde{u}^k)^T \gamma \varrho_k \beta_k F(\tilde{u}^k) \qquad \text{and} \qquad (u - u^{k+1})^T \gamma \varrho_k \beta_k F(\tilde{u}^k).$$

First, we deal with the term $(u^{k+1} - \tilde{u}^k)^T \gamma \varrho_k \beta_k F(\tilde{u}^k)$. Since $u^{k+1} \in \Omega$, substituting $u = u^{k+1}$ in (3.6a) we get

$$\begin{aligned} & (u^{k+1} - \tilde{u}^k)^T \gamma \varrho_k \beta_k F(\tilde{u}^k) \\ \geq \ \ & \gamma \varrho_k (u^{k+1} - \tilde{u}^k)^T d(u^k, \tilde{u}^k) \\ = \ \ & \gamma \varrho_k (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k) - \gamma \varrho_k (u^k - u^{k+1})^T d(u^k, \tilde{u}^k). \end{aligned} \tag{5.7}$$



To the first crossed term of the right hand side of (5.7), using (3.6b), we have

$$\gamma \varrho_k (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k) = \gamma \varrho_k^2 \|d(u^k, \tilde{u}^k)\|^2.$$

To the second crossed term of the right hand side of (5.7), using the Cauchy-Schwarz Inequality, we get

$$-\gamma \varrho_k (u^k - u^{k+1})^T d(u^k, \tilde{u}^k) \geq -\frac{1}{2}\|u^k - u^{k+1}\|^2 - \frac{1}{2}\gamma^2 \varrho_k^2 \|d(u^k, \tilde{u}^k)\|^2.$$

Substituting them in the right hand side of (5.7), we obtain

$$(u^{k+1} - \tilde{u}^k)^T \gamma \varrho_k \beta_k F(\tilde{u}^k) \geq \frac{1}{2}\gamma(2-\gamma)\varrho_k^2 \|d(u^k, \tilde{u}^k)\|^2 - \frac{1}{2}\|u^k - u^{k+1}\|^2. \tag{5.8}$$

Now, we turn to treat of the term $(u - u^{k+1})^T \gamma \varrho_k \beta_k F(\tilde{u}^k)$. Since $u^{k+1}$ is updated by (4.1b), $u^{k+1}$ is the projection of $\big(u^k - \gamma \varrho_k \beta_k F(\tilde{u}^k)\big)$ on $\Omega$, it follows from (2.1) that

$$\big\{\big(u^k - \gamma \varrho_k \beta_k F(\tilde{u}^k)\big) - u^{k+1}\big\}^T \big(u - u^{k+1}\big) \leq 0, \quad \forall\, u \in \Omega,$$

and consequently

$$\big(u - u^{k+1}\big)^T \gamma \varrho_k \beta_k F(\tilde{u}^k) \geq \big(u - u^{k+1}\big)^T \big(u^k - u^{k+1}\big), \quad \forall\, u \in \Omega.$$

Using the identity $a^T b = \frac{1}{2}\{\|a\|^2 - \|a-b\|^2 + \|b\|^2\}$ to the right hand side of the last inequality, we obtain

$$\big(u-u^{k+1}\big)^T\gamma\varrho_k\beta_kF(\tilde{u}^k) \ge \frac{1}{2}\big(\|u-u^{k+1}\|^2-\|u-u^k\|^2\big)+\frac{1}{2}\|u^k-u^{k+1}\|^2. \tag{5.9}$$

Adding (5.8) and (5.9) and using the definition of $q(\gamma)$, we get (5.6) and the proof is complete. $\square$

For the different projection and contraction methods, we have the same key inequality which is shown in Lemma 5.1 and Lemma 5.2, respectively. By setting $u=u^*$ in (5.3) and (5.6), we get

$$\|u^k-u^*\|^2-\|u^{k+1}-u^*\|^2 \ge 2\gamma\varrho_k\beta_k(\tilde{u}^k-u^*)^TF(\tilde{u}^k)+q(\gamma).$$

Because $(\tilde{u}^k-u^*)^TF(\tilde{u}^k)\ge(\tilde{u}^k-u^*)^TF(u^*)\ge 0$ and
$q(\gamma)=\gamma(2-\gamma)\varrho_k^2\|d(u^k,\tilde{u}^k)\|^2$, it follows from the last inequality that

$$\|u^{k+1}-u^*\|^2 \le \|u^k-u^*\|^2-\gamma(2-\gamma)\varrho_k^2\|d(u^k,\tilde{u}^k)\|^2.$$

This is just the form (4.11) in Section 4. In other words, the contraction property (4.11) of PC methods is the consequent result of Lemma 5.1 and Lemma 5.2, respectively.



For the convergence rate proof, we allow $\gamma\in(0,2]$. In this case, we still have $q(\gamma)\ge 0$. By using the monotonicity of $F$, from (5.3) and (5.6) we get

$$(u-\tilde{u}^k)^T\varrho_k\beta_kF(u)+\frac{1}{2\gamma}\|u-u^k\|^2 \ge \frac{1}{2\gamma}\|u-u^{k+1}\|^2,\quad \forall\, u\in\Omega. \tag{5.10}$$

This inequality is essential for the convergence rate proofs.

**Theorem 5.1** *For any integer $t>0$, we have a $\tilde{u}_t\in\Omega$ which satisfies*

$$(\tilde{u}_t-u)^TF(u)\le\frac{1}{2\gamma\Upsilon_t}\|u-u^0\|^2,\quad \forall u\in\Omega, \tag{5.11}$$

*where*

$$\tilde{u}_t=\frac{1}{\Upsilon_t}\sum_{k=0}^t\varrho_k\beta_k\tilde{u}^k \qquad \text{and} \qquad \Upsilon_t=\sum_{k=0}^t\varrho_k\beta_k. \tag{5.12}$$

**Proof**. Summing the inequality (5.10) over $k=0,\ldots,t$, we obtain

$$\Big(\big(\sum_{k=0}^t\varrho_k\beta_k\big)u-\sum_{k=0}^t\varrho_k\beta_k\tilde{u}^k\Big)^TF(u)+\frac{1}{2\gamma}\|u-u^0\|^2\ge 0,\quad \forall u\in\Omega.$$

Using the notations of $\Upsilon_t$ and $\tilde{u}_t$ in the above inequality, we derive

$$(\tilde{u}_t - u)^T F(u) \le \frac{\|u - u^0\|^2}{2\gamma\Upsilon_t}, \quad \forall u \in \Omega.$$

Indeed, $\tilde{u}_t \in \Omega$ because it is a convex combination of $\tilde{u}^0, \tilde{u}^1, \ldots, \tilde{u}^t$. The proof is complete. $\Box$

For given $u^k$, the predictor $\tilde{u}^k$ is given by (1.3a) and the prediction step size $\beta_k$ satisfies the condition (1.3b). Thus, the general conditions (3.5) are satisfied with $\varrho_k \ge c = \frac{1}{2}$. We choose (4.1a) (for the case that $u^k$ is not necessary in $\Omega$) or (4.1b) (for the case that $u^k \in \Omega$) to generate the new iterate $u^{k+1}$. Because $\varrho_k \ge \frac{1}{2}$, $\inf_{k\ge 0}\{\beta_k\} \ge \beta_{_L}$ and $\beta_{_L} = O(1/L)$, it follows from (5.12) that

$$\Upsilon_t \ge \frac{t+1}{2}\beta_{_L},$$

and thus the PC methods have $O(1/t)$ convergence rate. For any substantial set $\mathcal{D} \subset \Omega$, the PC methods reach

$$(\tilde{u}_t - u)^T F(u) \le \epsilon, \quad \forall\, u \in \mathcal{D}, \qquad \text{in at most} \qquad t = \left\lceil \frac{D^2}{\gamma \beta_{_L} \epsilon} \right\rceil$$



iterations, where $\tilde{u}_t$ is defined in (5.12) and $D = \sup\{\|u - u^0\| \,|\, u \in \mathcal{D}\}$. This convergence rate is in the ergodic sense, the statement (5.11) suggests us to take a larger parameter $\gamma \in (0, 2]$ in the correction steps of the PC methods.

# 6 Numerical experiments

This section is devoted to test the efficiency of the PC methods in comparison with the extragradient method [9]. Under the condition (1.3b), we have $\varrho_k > 1/2$. If we dynamically take $\gamma_k = 1/\varrho_k$ in (4.1b), then it becomes

$$u^{k+1} = P_\Omega[u^k - \beta_k F(\tilde{u}^k)], \tag{6.1}$$

which is the update form of the extragradient method [9]. Because $\gamma_k \varrho_k \equiv 1$, it follows from (5.10) that

$$(u - \tilde{u}^k)^T \beta_k F(u) + \frac{1}{2}\|u - u^k\|^2 \ge \frac{1}{2}\|u - u^{k+1}\|^2, \quad \forall u \in \Omega. \tag{6.2}$$

The results in Theorem 5.1 becomes

$$\tilde{u}_t = \frac{1}{\sum_{k=0}^{t} \beta_k} \sum_{k=0}^{t} \beta_k \tilde{u}^k \in \Omega,$$

and

$$(\tilde{u}_t - u)^T F(u) \le \frac{\|u - u^0\|^2}{2\big(\sum_{k=0}^{t} \beta_k\big)}, \qquad \forall u \in \Omega. \tag{6.3}$$

The $O(1/t)$ convergence rate follows from the above inequality directly. It should be mentioned that the projection-type method for VI$(\Omega, F)$ in [13] is a contraction method in the sense of $P$-norm, where $P$ is a positive definite matrix. In the Euclidean-norm, its update form is (4.1a).

**Test examples of nonlinear complementarity problems**.

We take nonlinear complementarity problems (NCP) as the test examples. The mapping $F(u)$ in the tested NCP is given by

$$F(u) = D(u) + Mu + q, \tag{6.4}$$

where $D(u) : \Re^n \to \Re^n$ is the nonlinear part, $M$ is an $n \times n$ matrix, and $q \in \Re^n$ is a vector.



- In $D(u)$, the nonlinear part of $F(u)$, the components are

  $$D_j(u) = d_j \cdot \arctan(a_j \cdot u_j),$$

  where $a$ and $d$ are random vectors[b] whose elements are in $(0, 1)$.
- The matrix $M$ in the linear part is given by $M = A^T A + B$. $A$ is an $n \times n$ matrix whose entries are randomly generated in the interval $(-5, +5)$, and $B$ is an $n \times n$ skew-symmetric random matrix ($B^T = -B$) whose entries[c] are in the interval $(-5, +5)$.

It is clear that the mapping composed in this way is monotone. We construct the following 3 sets of test examples by choosing different vector $q$ in (6.4).

1. In the first set of test examples, the elements of vector $q$ is generated from a uniform distribution in the interval $(-500, 500)$.

[b] A similar type of (small) problems was tested in [14] where the components of the nonlinear mapping $D(u)$ are $D_j(u) = c \cdot \arctan(u_j)$.

[c] In the paper by Harker and Pang [4], the matrix $M = A^T A + B + D$, where $A$ and $B$ are the same matrices as what we use here, and $D$ is a diagonal matrix with uniformly distributed random entries $d_{jj} \in (0.0, 0.3)$.

2. The second set[d] of test examples is similar to the first set. Instead of $q \in (-500, 500)$, the vector $q$ is generated from a uniform distribution in the interval $(-500, 0)$.

3. The third set of test examples has a known solution $u^* \in \Re^n_+$. Let vector $p$ be generated from a uniform distribution in the interval $(-10, 10)$ and

$$u^* = \max(p, 0). \tag{6.5}$$

By setting

$$w = \max(-p, 0) \quad \text{and} \quad q = w - (D(u^*) + Mu^*),$$

we have $F(u^*) = D(u^*) + Mu^* + q = w = \max(-p, 0)$. Thus,

$$(u^*)^T F(u^*) = \big(\max(p, 0)\big)^T \big(\max(-p, 0)\big) = 0.$$

In this way we constructed a test NCP with a known solution $u^*$ described in (6.5).

**Implementation details**.

[d]In [4], the similar problems in the first set are called easy problems while the 2-nd set problems are called hard problems.



For given $u^k$, we use (1.3) to produce $\tilde{u}^k$ with $\nu = 0.9$ in (1.3b). If $r_k := \beta_k \|F(u^k) - F(\tilde{u}^k)\| / \|u^k - \tilde{u}^k\|$ is too small, it will lead slow convergence. Therefore, if $r_k \le \mu = 0.3$, the trial parameter $\beta_k$ will be enlarged for the next iteration. These 'refined' strategies are necessary for fast convergence. The following is the implementation details.

Step 0. Set $\beta_0 = 1$, $u^0 \in \Omega$ and $k = 0$.
Step 1. $\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)]$,

$$r_k := \frac{\beta_k \|F(u^k) - F(\tilde{u}^k)\|}{\|u^k - \tilde{u}^k\|},$$

**while** $r_k > \nu$

$$\beta_k := 0.7 * \beta_k * \min\{1, \tfrac{1}{r_k}\}, \quad \tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)]$$

$$r_k := \frac{\beta_k \|F(u^k) - F(\tilde{u}^k)\|}{\|u^k - \tilde{u}^k\|},$$

**end(while)**

Use different forms ((6.1), (4.1a) or (4.1b)) to update $u^{k+1}$.

**If** $r_k \le \mu$ then $\beta_k := \beta_k * \nu * 0.9 / r_k$, **end(if)**

Step 2. $\beta_{k+1} = \beta_k$ and $k = k + 1$, go to Step 1.

The iterations begin with $u^0 = 0$, $\beta_0 = 1$ and stop as soon as

$$\frac{\|u^k - P_\Omega[u^k - F(u^k)]\|_\infty}{\|u^0 - P_\Omega[u^0 - F(u^0)]\|_\infty} \le 10^{-6}. \tag{6.6}$$

Since both $F(u^k)$ and $F(\tilde{u}^k)$ are involved in those methods recursions, each iteration of the test methods needs at least 2 times of evaluations of the mapping $F$. We use No. It and No. $F$ to denote the numbers of iterations and the evaluations of the mapping $F$, respectively. The size of the tested problems is from $500$ to $2000$. All codes are written in Matlab and run on a Lenovo X200 Computer with 2.53 GHz.

**Comparison beteeen the extragradient method and the PC method II**.

As mentioned in Section 4, replacing $\gamma\varrho_k$ in (4.1b) by 1, the PC method II becomes the extragradient method. According to the assertion in Theorem 4.1 and Theorem 5.1, we take the relaxation factor $\gamma = 2$ in the PC method II. The test results for the 3 sets of NCP are given in Tables 1-3, respectively.



Table 1. Numerical results of the first set examples

| Problem | Extra-Gradient Method $u^{k+1} = P_\Omega[u^k - \beta_k F(\tilde{u}^k)]$ | | | PC Method II $(\gamma = 2)$ $u^{k+1} = P_\Omega[u^k - 2\varrho_k\beta_k F(\tilde{u}^k)]$ | | |
|---|---|---|---|---|---|---|
| size $n$ | No. It | No. $F$ | CPU Sec | No. It | No. $F$ | CPU Sec |
| 500 | 496 | 1032 | 0.1626 | 224 | 490 | 0.0792 |
| 1000 | 439 | 917 | 1.5416 | 196 | 430 | 0.7285 |
| 2000 | 592 | 1236 | 7.8440 | 262 | 574 | 3.7305 |

Table 2. Numerical results of the second set examples

| Problem | Extra-Gradient Method $u^{k+1} = P_\Omega[u^k - \beta_k F(\tilde{u}^k)]$ | | | PC Method II $(\gamma = 2)$ $u^{k+1} = P_\Omega[u^k - 2\varrho_k\beta_k F(\tilde{u}^k)]$ | | |
|---|---|---|---|---|---|---|
| size $n$ | No. It | No. $F$ | CPU Sec | No. It | No. $F$ | CPU Sec |
| 500 | 1157 | 2412 | 0.3921 | 510 | 1113 | 0.1938 |
| 1000 | 1197 | 2475 | 4.1946 | 533 | 1162 | 1.9350 |
| 2000 | 1487 | 3099 | 19.6591 | 669 | 1452 | 9.3591 |

Table 3. Numerical results of the third set examples

| Problem size $n$ | Extra-Gradient Method $u^{k+1} = P_{\Omega}[u^k - \beta_k F(\tilde{u}^k)]$ | | | PC Method II $(\gamma = 2)$ $u^{k+1} = P_{\Omega}[u^k - 2\varrho_k \beta_k F(\tilde{u}^k)]$ | | |
|---|---|---|---|---|---|---|
| | No. It | No. $F$ | CPU Sec | No. It | No. $F$ | CPU Sec |
| 500 | 633 | 1318 | 0.2109 | 279 | 610 | 0.0988 |
| 1000 | 700 | 1458 | 2.4544 | 308 | 673 | 1.1272 |
| 2000 | 789 | 1643 | 10.4436 | 346 | 756 | 4.8455 |

In the third test examples, as the stop criterium is satisfied, we have $\|u^k - u^*\|_\infty \approx 2 \times 10^{-4}$ by using the both test methods. The PC Method II and the extragradient method use the same direction but different step size in the correction step. The numerical results show that the PC method II is much efficient than the extragradient method. Even if the PC methods need to calculate the step size $\varrho_k$ in each iteration, while the computational load required by the additional effort is significantly less than the dominating task (the evaluations of $F(u^k)$ and $F(\tilde{u}^k)$). It is observed that

$$\frac{\text{Computational load of PC Method II}}{\text{Computational load of the extragradient method}} < 50\%.$$



**Comparison between PC method I and PC method II**.

The different PC methods use the one of the geminate directions but the same step size in their correction forms. In order to ensure $\vartheta_I(\gamma) > 0$, we take $\gamma = 1.9$ in (4.1) for the both update forms. The test results for the 3 sets of NCP are given in Tables 4-6, respectively.

Table 4. Numerical results of the first set examples

| Problem size $n$ | PC Method I $(\gamma = 1.9)$ $u^{k+1} = u^k - \gamma\varrho_k d(u^k, \tilde{u}^k)$ | | | PC Method II $(\gamma = 1.9)$ $u^{k+1} = P_{\Omega}[u^k - \gamma\varrho_k \beta_k F(\tilde{u}^k)]$ | | |
|---|---|---|---|---|---|---|
| | No. It | No. $F$ | CPU Sec | No. It | No. $F$ | CPU Sec |
| 500 | 294 | 625 | 0.1060 | 233 | 507 | 0.0885 |
| 1000 | 253 | 546 | 0.9451 | 204 | 445 | 0.7714 |
| 2000 | 334 | 704 | 4.5035 | 271 | 591 | 3.7896 |

Table 5. Numerical results of the second set examples

| Problem size $n$ | PC Method I $(\gamma = 1.9)$ $u^{k+1} = u^k - \gamma\varrho_k d(u^k, \tilde{u}^k)$ | | | PC Method II $(\gamma = 1.9)$ $u^{k+1} = P_\Omega[u^k - \gamma\varrho_k\beta_k F(\tilde{u}^k)]$ | | |
|---|---|---|---|---|---|---|
| | No. It | No. $F$ | CPU Sec | No. It | No. $F$ | CPU Sec |
| 500 | 594 | 1273 | 0.2192 | 539 | 1170 | 0.2014 |
| 1000 | 635 | 1345 | 2.3151 | 559 | 1213 | 2.0908 |
| 2000 | 772 | 1641 | 10.4909 | 701 | 1518 | 9.7359 |

Table 6. Numerical results of the third set examples

| Problem size $n$ | PC Method I $(\gamma = 1.9)$ $u^{k+1} = u^k - \gamma\varrho_k d(u^k, \tilde{u}^k)$ | | | PC Method II $(\gamma = 1.9)$ $u^{k+1} = P_\Omega[u^k - \gamma\varrho_k\beta_k F(\tilde{u}^k)]$ | | |
|---|---|---|---|---|---|---|
| | No. It | No. $F$ | CPU Sec | No. It | No. $F$ | CPU Sec |
| 500 | 348 | 741 | 0.1328 | 295 | 642 | 0.1162 |
| 1000 | 368 | 782 | 1.3584 | 328 | 713 | 1.2441 |
| 2000 | 423 | 900 | 5.7394 | 370 | 803 | 5.1350 |

Between the PC methods, PC method II needs fewer iterations than PC method I, this



evidence coincides with the assertions in Theorem 4.1 (see (4.3) and (4.4)). Thus, we suggest to use PC method II when the projection on $\Omega$ is easy to be carried out. Otherwise (when the projection is the dominating task in the iteration), we use PC method I because its update form (4.1a) does not contain the projection.

# 7 Conclusions

In a unified framework, we proved the $O(1/t)$ convergence rate of the projection and contraction methods for monotone variational inequalities. The convergence rate is the same as that for the extragradient method. In fact, our convergence rate include the extragradient method as a special case. The complexity analysis in this paper is based on the general conditions (3.5) and thus can be extended to a broaden class of similar contraction methods. Preliminary numerical results indicate that the PC methods do outperform the extragradient method.

**注记** 我们用 $P_\Omega(\cdot)$ 表示欧氏范数下在凸集 $\Omega$ 上的投影. 也就是说, 对给定的 $v$,

$$P_\Omega(v) = \arg\min\{\|u - v\| \mid u \in \Omega\}.$$

设 $\Omega \subset R^n$ 是闭凸集, 则对任意的 $v \in \Re^n$, 有

$$(v - P_\Omega(v))^T(u - P_\Omega(v)) \le 0, \quad \forall u \in \Omega. \tag{7.1}$$

在求解单调变分不等式的投影收缩算法中, 对给定的当前点 $u^k$ 和 $\beta_k > 0$, 我们利用投影

$$\tilde{u}^k = P_\Omega[u^k - \beta_k F(u^k)], \tag{7.2}$$

生成一个预测点 $\tilde{u}^k$. 在投影的基本性质 (7.1) 中, 令 $v = u^k - \beta_k F(u^k)$, 根据 (7.2), $\tilde{u}^k = P_\Omega(v)$, 就有

$$\tilde{u}^k \in \Omega, \quad \{[u^k - \beta_k F(u^k)] - \tilde{u}^k\}^T(u - \tilde{u}^k) \le 0, \quad \forall u \in \Omega.$$

进而得到

$$\tilde{u}^k \in \Omega, \quad (u - \tilde{u}^k)^T \beta_k F(u^k) \ge (u - \tilde{u}^k)^T(u^k - \tilde{u}^k), \quad \forall\, u \in \Omega.$$



两边都加上 $\quad -\beta_k[F(u^k) - F(\tilde{u}^k)], \quad$ 就有

$$\tilde{u}^k \in \Omega, \ \ (u - \tilde{u}^k)^T \beta_k F(\tilde{u}^k) \ge (u - \tilde{u}^k)^T d(u^k, \tilde{u}^k), \ \ \forall u \in \Omega, \tag{7.3}$$

其中

$$d(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k) - \beta_k[F(u^k) - F(\tilde{u}^k)] \tag{7.4}$$

这种我们希望的形式. (7.3) 表明 $\tilde{u}^k$ 是一个特定的变分不等式的解, 它就是 (3.6a), 或者说 (3.5a) 的等价形式.

我们已经并且会进一步看到, 对由凸优化来的变分不等式

$$w^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \ge 0, \quad \forall w \in \Omega, \tag{7.5}$$

通过求解一些子问题, 产生的预测点 $\tilde{w}^k$ 是与 (7.3) 类似的变分不等式

$$\tilde{w}^k \in \Omega, \ \ \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \ge (w - \tilde{w}^k)^T H(w^k - \tilde{w}^k), \ \ \forall\, w \in \Omega,$$

或者

$$\tilde{w}^k \in \Omega, \ \ \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \ge (w - \tilde{w}^k)^T HM(w^k - \tilde{w}^k), \ \ \forall\, w \in \Omega,$$

的解. 这是我们能把求变分不等式的方法移植到结构型凸优化的根本原因.

# References

[1] E. Blum and W. Oettli, *Mathematische Optimierung: Grundlagen und Verfahren. Ökonometrie und Unternehmensforschung*. Berlin-Heidelberg-New York, Springer-Verlag, 1975.

[2] X.J. Cai, G.Y. Gu and B. S. He, On the O(1/t) convergence rate of the projection and contraction methods for variational inequalities with Lipschitz continuous monotone operators, *Comput. Optim. Appl.,* **57**, pp. 339-363, 2013.

[3] F. Facchinei and J.S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems, Vol. I and II*. Springer Series in Operations Research. Springer Verlag, New York, 2003.

[4] P.T. Harker and J.S. Pang, A damped-Newton method for the linear complementarity problem, *Lectures in Applied Mathematics* **26**, pp. 265–284, 1990.

[5] B.S. He, A class of projection and contraction methods for monotone variational inequalities, *Applied Mathematics and optimization* **35**, pp. 69–76, 1997.

[6] B.S. He and L.-Z Liao, Improvements of some projection methods for monotone nonlinear variational inequalities, *Journal of Optimization Theory and Applications* **112**, pp. 111-128, 2002.

[7] B.S. He and M.-H. Xu, A general framework of contraction methods for monotone variational inequalities, *Pacific J. Optimization* **4**, pp. 195-212, 2008.

[8] B.S. He, X.M. Yuan and J.J.Z. Zhang, Comparison of two kinds of prediction-correction methods for monotone variational inequalities, *Computational Optimization and Applications* **27**, pp. 247-267, 2004.



[9] G.M. Korpelevich, The extragradient method for finding saddle points and other problems, *Ekonomika i Matematicheskie Metody* **12**, pp. 747-756, 1976.

[10] A. Nemirovski, Prox-method with rate of convergence $O(1/t)$ for variational inequality with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems, *SIAM J. Optimization* **15**, pp. 229-251, 2005.

[11] Y. Pan and L. Pavel, Games with coupled propagated constraints in optical networks with multi-link topologies, *Automatica* **45**, pp. 871-880, 2009.

[12] F. Sha, Large Margin Training of Acoustic Models for Speech Recognition, PhD Thesis, Computer and Information Science, University of Pennsylvania, 2007.

[13] M.V. Solodov and P. Tseng, Modified projection-type methods for monotone variational inequalities, *SIAM J. Control and Optimization* **34**, pp. 1814–1830, 1996.

[14] K. Taji, M. Fukushima and I. Ibaraki, A globally convergent Newton mmethod for solving strongly monotone variational inequalities. *Mathematical Programming* **58**, pp. 369-383, 1993.

[15] B. Taskar, S. Lacoste-Julien and M. I. Jordan. Structured prediction via Extragradient method, In Y. Weiss and B. Schoelkopf and J. Platt (Eds.), *Advances in Neural Information Processing Systems (NIPS)* **18**, 2006.

[16] P. Tseng, On accelerated proximal gradient methods for convex-concave optimization, Department of Mathematics, University of Washington, Seattle, WA 98195, USA, 2008.

# 凸优化和单调变分不等式的收缩算法

## 第廿讲: 交替方向法的收敛速率

On the $O(1/t)$ convergence rate of
the alternating directions method of multipliers

南京大学数学系 何炳生
hebma@nju.edu.cn

The context of this lecture is based on the paper [13, 15]



## 1 Structured constrained convex optimization

We consider the following structured constrained convex optimization problem

$$\min \{\theta_1(x) + \theta_2(y) \mid Ax + By = b,\ x \in \mathcal{X},\ y \in \mathcal{Y}\} \quad (1.1)$$

where $\theta_1(x) : \Re^{n_1} \to \Re,\ \theta_2(y) : \Re^{n_2} \to \Re$ are convex functions (but not necessary smooth), $A \in \Re^{m\times n_1}$, $B \in \Re^{m\times n_2}$ and $b \in \Re^m$, $\mathcal{X} \subset \Re^{n_1}$, $\mathcal{Y} \subset \Re^{n_2}$ are given closed convex sets. Let $n = n_1 + n_2$.

The task of solving the problem (1.1) is to find an $(x^*, y^*, \lambda^*) \in \Omega$, such that

$$\begin{cases} \theta_1(x) - \theta_1(x^*) + (x - x^*)^T(-A^T\lambda^*) \ge 0, \\ \theta_2(y) - \theta_2(y^*) + (y - y^*)^T(-B^T\lambda^*) \ge 0, \quad \forall\, (x, y, \lambda) \in \Omega, \\ (\lambda - \lambda^*)^T(Ax^* + By^* - b) \ge 0, \end{cases} \quad (1.2)$$

where

$$\Omega = \mathcal{X} \times \mathcal{Y} \times \Re^m.$$

By denoting

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

and

$$\theta(u) = \theta_1(x) + \theta_2(y),$$

the first order optimal condition (1.2) can be written in a compact form such as

$$\boxed{\textbf{VI}(\Omega, F, \theta) \;\; w^* \in \Omega, \;\; \theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \geq 0, \;\; \forall\, w \in \Omega. \quad (1.3)}$$

Note that the mapping $F$ is monotone. We use $\Omega^*$ to denote the solution set of the variational inequality (1.3). We define some matrices which will greatly simplify the



notations in our analysis. More specifically, let

$$H_0 = \begin{pmatrix} 0 & 0 \\ \beta B^T B & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix}, \quad Q_0 = \begin{pmatrix} 0 & 0 \\ \beta B^T B & 0 \\ -B & \frac{1}{\beta} I_m \end{pmatrix}, \tag{1.4}$$

$$H = \begin{pmatrix} \beta B^T B & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix}, \quad Q = \begin{pmatrix} \beta B^T B & 0 \\ -B & \frac{1}{\beta} I_m \end{pmatrix}, \tag{1.5}$$

and

$$M = \begin{pmatrix} I_{n_2} & 0 \\ -\beta B & I_m \end{pmatrix}. \tag{1.6}$$

For the matrices $H$, $Q$ and $M$ be defined in (1.5) and (1.6), we have

$$Q = HM, \tag{1.7}$$

In addition, because

$$
\begin{aligned}
(Q^T + Q) - M^T H M &= M^T H + H M - M^T H M \\
&= H - (M^T - I) H (M - I) \\
&= \begin{pmatrix} \beta B^T B & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \\
&\quad - \begin{pmatrix} 0 & -\beta B^T \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \beta B^T B & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} 0 & 0 \\ -\beta B & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \succeq 0.
\end{aligned}
$$

we have

$$(Q^T + Q) - M^T H M \succeq 0. \tag{1.8}$$

These relations will simplify the convergence rate proofs in this lecture.



# 2 Alternating Direction Method

**Applied ADM to the structure VI** $(y^k, \lambda^k) \Rightarrow (y^{k+1}, \lambda^{k+1})$

1. First, for given $(y^k, \lambda^k)$, $x^{k+1}$ is the solution of the following problem

$$x^{k+1} = \text{Argmin}\{\theta_1(x) + \frac{\beta}{2}\|(Ax + By^k - b) - \frac{1}{\beta}\lambda^k\|^2 | x \in \mathcal{X}\} \tag{2.1a}$$

2. Use $\lambda^k$ and the obtained $x^{k+1}$, $y^{k+1}$ is the solution of the following problem

$$y^{k+1} = \text{Argmin}\{\theta_2(y) + \frac{\beta}{2}\|(Ax^{k+1} + By - b) - \frac{1}{\beta}\lambda^k\|^2 | y \in \mathcal{Y}\} \tag{2.1b}$$

3. Update the

$$\lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b). \tag{2.1c}$$

As mentioned in [1], the variable $x$ is an intermediate variable during the ADM iterations since it essentially requires only $(y^k, \lambda^k)$ to generate the $(k+1)$-th iterate. For this reason, in the following analysis, we sometimes use the notations $v^k = (y^k, \lambda^k)$ and

$\mathcal{V} = \mathcal{Y} \times \Re^m$, and we let

$$\mathcal{V}^* := \{v^* = (y^*, \lambda^*) \,|\, w^* = (x^*, y^*, \lambda^*) \in \Omega^*\}.$$

**Lemma 2.1** *Let $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1}) \in \Omega$ be generated by* (2.1) *from the given $v^k = (y^k, \lambda^k)$. Then, we have*

$$\theta(u) - \theta(u^{k+1}) + (w - w^{k+1})^T \Big\{ F(w^{k+1}) + \eta(y^k, y^{k+1}) + H_0(v^{k+1} - v^k) \Big\} \geq 0, \quad \forall\, w \in \Omega. \tag{2.2}$$

*where*

$$\eta(y^k, y^{k+1}) = \beta \begin{pmatrix} A^T \\ B^T \\ 0 \end{pmatrix} B(y^k - y^{k+1}) \tag{2.3}$$

*and the matrix $H_0$ is defined in* (1.4).

**Proof**. Note that the solution of (2.1a) satisfies

$$x^{k+1} \in \mathcal{X}, \quad \theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^T \big\{ -A^T\lambda^k + \beta A^T\big(Ax^{k+1} + By^k - b\big)\big\} \geq 0, \ \forall\, x \in \mathcal{X}. \tag{2.4a}$$



And the solution of (2.1b) satisfies

$$y^{k+1} \in \mathcal{Y}, \quad \theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^T \big\{ -B^T\lambda^k + \beta B^T\big(Ax^{k+1} + By^{k+1} - b\big)\big\} \geq 0, \ \forall\, y \in \mathcal{Y}. \tag{2.4b}$$

Substituting $\lambda^{k+1}$ (see (2.1c)) in (2.4) (eliminating $\lambda^k$ in (2.4)), we get

$$x^{k+1} \in \mathcal{X}, \quad \theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^T \big\{ -A^T\lambda^{k+1} + \beta A^T B(y^k - y^{k+1})\big\} \geq 0, \ \forall\, x \in \mathcal{X}. \tag{2.5a}$$

and

$$y^{k+1} \in \mathcal{Y}, \quad \theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^T \{ -B^T\lambda^{k+1}\} \geq 0, \ \forall\, y \in \mathcal{Y}. \tag{2.5b}$$

For analysis convenience, we rewrite (2.5) as $u^{k+1} = (x^{k+1}, y^{k+1}) \in \mathcal{X} \times \mathcal{Y}$.

$$\theta(u) - \theta(u^{k+1}) + \begin{pmatrix} x - x^{k+1} \\ y - y^{k+1} \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\lambda^{k+1} \\ -B^T\lambda^{k+1} \end{pmatrix} + \beta \begin{pmatrix} A^T \\ B^T \end{pmatrix} B(y^k - y^{k+1}) \right.$$
$$\left. + \begin{pmatrix} 0 & 0 \\ 0 & \beta B^T B \end{pmatrix} \begin{pmatrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{pmatrix} \right\} \geq 0, \ \forall (x,y) \in \mathcal{X} \times \mathcal{Y}.$$

Combining the last inequality with (2.1c), we have $w^{k+1} \in \Omega$ and

$$\theta(u) - \theta(u^{k+1}) + \begin{pmatrix} x - x^{k+1} \\ y - y^{k+1} \\ \lambda - \lambda^{k+1} \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\lambda^{k+1} \\ -B^T\lambda^{k+1} \\ Ax^{k+1} + By^{k+1} - b \end{pmatrix} \right.$$
$$\left. + \beta \begin{pmatrix} A^T \\ B^T \\ 0 \end{pmatrix} B(y^k - y^{k+1}) + \begin{pmatrix} 0 & 0 \\ \beta B^T B & 0 \\ 0 & \frac{1}{\beta} I_m \end{pmatrix} \begin{pmatrix} y^{k+1} - y^k \\ \lambda^{k+1} - \lambda^k \end{pmatrix} \right\} \geq 0, \quad (2.6)$$

for any $w \in \Omega$. The last inequality can be written as a compact form of (2.2). $\square$



# 3 Contractive property of ADM

Contractive property means that the sequence $\{\|v^k - v^*\|_H^2\}$ in ADM is monotonically deceasing. Based on the analysis in the last section, we have the following lemma.

**Lemma 3.1** *Let* $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1}) \in \Omega$ *be generated by* (2.1) *from the given* $v^k = (y^k, \lambda^k)$. *Then, we have*

$$(v^{k+1} - v^*)^T H(v^k - v^{k+1}) \geq (w^{k+1} - w^*)^T \eta(y^k, y^{k+1}), \quad (3.1)$$

*and the matrix* $H$ *is defined in* (1.5) *and* $\eta(y^k, y^{k+1})$ *is defined in* (2.3).

**Proof**. Setting $w = w^*$ in (2.2), and using the structures of $H_0$ and $H$, we get

$$\begin{aligned} (v^{k+1} - v^*)^T H(v^k - v^{k+1}) &\geq (w^{k+1} - w^*)^T \eta(y^k, y^{k+1}) \\ &\quad + \big(\theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^T F(w^{k+1})\big). \end{aligned} \quad (3.2)$$

Since $F$ is monotone, it follows that

$$\begin{aligned} &\theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^T F(w^{k+1}) \\ \geq \quad &\theta(u^{k+1}) - \theta(u^*) + (w^{k+1} - w^*)^T F(w^*) \geq 0. \end{aligned}$$

The last inequality is due to $w^{k+1} \in \Omega$ and $w^* \in \Omega^*$ (see (1.3)). Substituting it in (3.2), the lemma is proved. $\Box$

**Lemma 3.2** *Let* $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1}) \in \Omega$ *be generated by* (2.1) *from the given* $v^k = (y^k, \lambda^k)$ *and the vector* $\eta(y^k, y^{k+1})$ *be defined in* (2.3)*. Then, we have*

$$(w^{k+1} - w^*)^T \eta(y^k, y^{k+1}) = (\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}), \tag{3.3}$$

*and*

$$(\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}) \geq 0. \tag{3.4}$$

**Proof**. By using $\eta(y^k, y^{k+1})$ (see (2.3)), $Ax^* + By^* = b$ and (2.1c), we have

$$\begin{aligned} & (w^{k+1} - w^*)^T \eta(y^k, y^{k+1}) \\ = \quad & (B(y^k - y^{k+1}))^T \beta\{(Ax^{k+1} + By^{k+1}) - (Ax^* + By^*)\} \\ = \quad & (B(y^k - y^{k+1}))^T \beta(Ax^{k+1} + By^{k+1} - b) \\ = \quad & (\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}). \end{aligned}$$

Because (2.5b) is true for the $k$-th iteration and the previous iteration, we have

$$\theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^T \{-B^T \lambda^{k+1}\} \geq 0, \ \forall\, y \in \mathcal{Y}, \tag{3.5}$$



and

$$\theta_2(y) - \theta_2(y^k) + (y - y^k)^T \{-B^T \lambda^k\} \geq 0, \ \forall\, y \in \mathcal{Y}, \tag{3.6}$$

Setting $y = y^k$ in (3.5) and $y = y^{k+1}$ in (3.6), respectively, and then adding the two resulting inequalities, we get

$$(\lambda^k - \lambda^{k+1})^T B(y^k - y^{k+1}) \geq 0.$$

The assertions of this lemma are proved. $\Box$

**Lemma 3.3** *Let* $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1}) \in \Omega$ *be generated by* (2.1) *from the given* $v^k = (y^k, \lambda^k)$*. Then, we have*

$$(v^{k+1} - v^*)^T H(v^k - v^{k+1}) \geq 0, \quad \forall\, v^* \in \mathcal{V}^*. \tag{3.7}$$

**Proof**. The assertion follows (3.1), (3.3) and (3.4) directly. $\Box$

Even though $H$ is positive semi-definite (see (1.6) when $B$ is not full column rank), in this lecture we use $\|v - \tilde{v}\|_H$ to denote that

$$\|v - \tilde{v}\|_H^2 = (v - \tilde{v})^T H(v - \tilde{v}) = \beta\|B(y - \tilde{y})\|^2 + \frac{1}{\beta}\|\lambda - \tilde{\lambda}\|^2.$$

Based on the above mentioned lemmas, the contractive property of the sequence $\{\|v^k - v^*\|_H\}$ follows directly.

**Theorem 3.1** *Let* $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1}) \in \Omega$ *be generated by* (2.1) *from the given* $v^k = (y^k, \lambda^k)$. *Then, we have*

$$\|v^{k+1} - v^*\|_H^2 \le \|v^k - v^*\|_H^2 - \|v^k - v^{k+1}\|_H^2, \ \forall\, v^* \in \mathcal{V}^*. \tag{3.8}$$

**Proof**. By using (3.7), we have

$$\begin{aligned} \|v^k - v^*\|_H^2 &= \|(v^{k+1} - v^*) + (v^k - v^{k+1})\|_H^2 \\ &= \|v^{k+1} - v^*\|_H^2 + 2(v^{k+1} - v^*)^T H(v^k - v^{k+1}) \\ &\quad + \|v^k - v^{k+1}\|_H^2 \\ &\ge \|v^{k+1} - v^*\|_H^2 + \|v^k - v^{k+1}\|_H^2, \end{aligned}$$

and thus (3.8) is proved. □

The inequality (3.8) in Theorem 3.1 demonstrates the contractive property of the ADM.



# 4 Defining an associated sequence $\{\tilde{w}^k\}$

For the convergence rate proof we preferably define an associated sequence $\{\tilde{w}^k\}$ by

$$\tilde{w}^k = \begin{pmatrix} \tilde{x}^k \\ \tilde{y}^k \\ \tilde{\lambda}^k \end{pmatrix} = \begin{pmatrix} x^{k+1} \\ y^{k+1} \\ \lambda^k - \beta(Ax^{k+1} + By^k - b) \end{pmatrix}, \tag{4.1}$$

where $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1})$ is generated by the ADM (2.1). Note that only the $\lambda$-part of $\tilde{w}^k$ and $w^{k+1}$ is different. By using (4.1) and (2.1c), we obtain the following useful relationship

$$\begin{pmatrix} y^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ \lambda^k \end{pmatrix} - \begin{pmatrix} I_{n_2} & 0 \\ -\beta B & I_m \end{pmatrix} \begin{pmatrix} y^k - \tilde{y}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}, \tag{4.2}$$

which can be rewritten into a compact form:

$$v^{k+1} = v^k - M(v^k - \tilde{v}^k), \tag{4.3}$$

where the matrix $M$ is defined in (1.6). These notations simplify our presentation much.

Now, we start to prove some properties of the sequence $\{\tilde{w}^k\}$. The first lemma quantifies the discrepancy between the point $\tilde{w}^k$ and a solution point of (1.3).

**Lemma 4.1** *Let $\{v^k\}$ be the sequence generated by (2.1) and the associated sequence $\{\tilde{w}^k\}$ be defined by (4.1). Then, we have*

$$\tilde{w}^k \in \Omega,\ \theta(u)-\theta(\tilde{u}^k)+(w-\tilde{w}^k)^T\{F(\tilde{w}^k)+Q_0(\tilde{v}^k-v^k)\} \geq 0,\ \forall\, w \in \Omega, \quad (4.4)$$

*where $Q_0$ is defined in* (1.4)*.*

**Proof**. The assertion is followed from Lemma 2.1. To see this, we observe the terms

$$F(w^{k+1}) + \eta(y^k, y^{k+1}) \qquad \text{and} \qquad H_0(v^{k+1} - v^k).$$

in (2.2). Note that (4.1) implies

$$x^{k+1} = \tilde{x}^k, \qquad y^{k+1} = \tilde{y}^k, \tag{4.5a}$$

and

$$\lambda^{k+1} = \tilde{\lambda}^k + \beta B(y^k - y^{k+1}). \tag{4.5b}$$



According to the above relations, we have

$$\begin{pmatrix} -A^T\lambda^{k+1} \\ -B^T\lambda^{k+1} \\ Ax^{k+1} + By^{k+1} - b \end{pmatrix} + \beta \begin{pmatrix} A^T \\ B^T \\ 0 \end{pmatrix} B(y^k - y^{k+1}) = \begin{pmatrix} -A^T\tilde{\lambda}^k \\ -B^T\tilde{\lambda}^k \\ A\tilde{x}^k + B\tilde{y}^k - b \end{pmatrix},$$

and thus

$$F(w^{k+1}) + \eta(y^k, y^{k+1}) = F(\tilde{w}^k). \tag{4.6}$$

Again, it follows from (4.5) that

$$\frac{1}{\beta}(\lambda^{k+1} - \lambda^k) = -B(\tilde{y}^k - y^k) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k),$$

and consequently

$$\begin{pmatrix} 0 & 0 \\ \beta B^T B & 0 \\ 0 & \frac{1}{\beta}I_m \end{pmatrix} \begin{pmatrix} y^{k+1} - y^k \\ \lambda^{k+1} - \lambda^k \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ \beta B^T B & 0 \\ -B & \frac{1}{\beta}I_m \end{pmatrix} \begin{pmatrix} \tilde{y}^k - y^k \\ \tilde{\lambda}^k - \lambda^k \end{pmatrix}.$$

By using the matrices $H_0$ and $Q_0$ (see (1.4)), the last equation can be written as means that

$$H_0(v^{k+1} - v^k) = Q_0(\tilde{v}^k - v^k). \tag{4.7}$$

Substituting (4.6) and (4.7) in (2.2), the assertion is proved. $\square$

# 5 Convergence in an ergodic sense

The reason inspiring us to investigate the convergence rate of ADM via the VI approach, rather than the conventional approaches based on the functional values in the literature, is mainly due to the Theorem 2.3.5 in [5] which provides an insightful characterization for the solution set of a generic VI. In the following theorem, we specific the Theorem 2.3.5 in [5] for the particular VI$(\Omega, F, \theta)$ under consideration. The proof is an incremental extension of Theorem 2.3.5 in [5]. But, we include all the details for completeness.

**Characterizing of the solution set**

**Theorem 5.1** *The solution set of VI$(\Omega, F, \theta)$ is convex and it can be characterized as*

$$\Omega^* = \bigcap_{w\in\Omega} \big\{\tilde{w} \in \Omega : \big(\theta(u) - \theta(\tilde{u})\big) + (w - \tilde{w})^T F(w) \geq 0\big\}. \tag{5.1}$$



**Proof**. Indeed, if $\tilde{w} \in \Omega^*$, we have

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

By using the monotonicity of $F$ on $\Omega$, this implies

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

Thus, $\tilde{w}$ belongs to the right-hand set in (5.1). Conversely, suppose $\tilde{w}$ belongs to the latter set. Let $w \in \Omega$ be arbitrary. The vector

$$\bar{w} = \alpha\tilde{w} + (1-\alpha)w$$

belongs to $\Omega$ for all $\alpha \in (0, 1)$. Thus we have

$$\theta(\bar{u}) - \theta(\tilde{u}) + (\bar{w} - \tilde{w})^T F(\bar{w}) \geq 0. \tag{5.2}$$

Because $\theta(\cdot)$ is convex, we have

$$\theta(\bar{u}) \leq \alpha\theta(\tilde{u}) + (1-\alpha)\theta(u).$$

Substituting it in (5.2), we get

$$(\theta(u) - \theta(\tilde{u})) + (w - \tilde{w})^T F(\alpha\tilde{w} + (1-\alpha)w) \geq 0$$

for all $\alpha \in (0,1)$. Letting $\alpha \to 1$ yields

$$(\theta(u) - \theta(\tilde{u})) + (w - \tilde{w})^T F(\tilde{w}) \geq 0.$$

Thus $\tilde{w} \in \Omega^*$. Now, we turn to prove the convexity of $\Omega^*$. For each fixed but arbitrary $w \in \Omega$, the set

$$\{\tilde{w} \in \Omega : \ \theta(\tilde{u}) + \tilde{w}^T F(w) \leq \theta(u) + w^T F(w)\}$$

and its equivalent expression

$$\{\tilde{w} \in \Omega : \ \big(\theta(u) - \theta(\tilde{u})\big) + (w - \tilde{w})^T F(w) \geq 0\}$$

is convex. Since the intersection of any number of convex sets is convex, it follows that the solution set of VI$(\Omega, F, \theta)$ is convex. $\Box$

Theorem 5.1 thus implies that $\tilde{w} \in \Omega$ is an approximate solution of VI$(\Omega, F, \theta)$ with the accuracy $\epsilon > 0$ if it satisfies

$$\theta(u) - \theta(\tilde{u}) + F(w)^T (w - \tilde{w}) \geq -\epsilon, \ \ \forall w \in \Omega_{\tilde{w}},$$

where

$$\Omega_{\tilde{w}} = \{w \in \Omega \mid \|w - \tilde{w}\| \leq 1\}.$$



In the rest, our purpose is to show that after $t$ iterations of the ADM (2.1), we can find $\tilde{w} \in \Omega$ such that

$$\theta(\tilde{u}) - \theta(u) + (\tilde{w} - w)^T F(w) \leq \epsilon, \ \ \forall w \in \Omega_{\tilde{w}} \tag{5.3}$$

with $\epsilon = O(1/t)$. The convergence rate $O(1/t)$ of the ADM (2.1) is thus proved.

Since $H$ is symmetric and positive semi-definite, for notational convenience we use $\|v - \tilde{v}\|_H$ to denote

$$\|v - \tilde{v}\|_H = \big((v - \tilde{v})^T H (v - \tilde{v})\big)^{1/2}.$$

Together with the notations of $H$ and $Q$, it is trivial to verify that (4.4) can be written into

$$\big(\theta(u) - \theta(\tilde{u}^k)\big) + (w - \tilde{w}^k)^T F(w) \geq (v - \tilde{v}^k)^T HM(v^k - \tilde{v}^k), \quad \forall\, w \in \Omega, \tag{5.4}$$

and we omit the proof.

Now, we deal with the right-hand side of (5.4), and we want to find a lower bound of it in terms of $\|v - v^k\|_H^2$ and $\|v - v^{k+1}\|_H^2$. This is realized in the following lemma.

**Lemma 5.1** *Let $\{v^k\}$ be the sequence generated by the ADM (2.1) and the associated sequence $\{\tilde{w}^k\}$ be defined by (4.1). Then we have*

$$(v - \tilde{v}^k)^T HM(v^k - \tilde{v}^k) \geq \frac{1}{2}\big(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2\big), \quad \forall v \in \mathcal{V}. \tag{5.5}$$

**Proof**. First, by using $M(v^k - \tilde{v}^k) = (v^k - v^{k+1})$ (see (4.3)), it follows that

$$(v - \tilde{v}^k)^T HM(v^k - \tilde{v}^k) = (v - \tilde{v}^k)^T H(v^k - v^{k+1}).$$

Therefore, in order to obtain (5.5) we need only to prove that

$$(v - \tilde{v}^k)^T H(v^k - v^{k+1}) \geq \frac{1}{2}\big(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2\big), \quad \forall v \in \mathcal{V}. \tag{5.6}$$

Applying the identity

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

with

$$a = v, \quad b = \tilde{v}^k, \quad c = v^k \quad \text{and} \quad d = v^{k+1},$$



we thus get

$$\begin{aligned}(v - \tilde{v}^k)^T H(v^k - v^{k+1}) \\ = \quad \frac{1}{2}\big(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2\big) + \frac{1}{2}\big(\|v^k - \tilde{v}^k\|_H^2 - \frac{1}{2}\|v^{k+1} - \tilde{v}^k\|_H^2\big).\end{aligned}$$

To show (5.6), we need only to demonstrate that

$$\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2 \geq 0. \tag{5.7}$$

Recall (2.1) and (4.1), we then get

$$\|v^{k+1} - \tilde{v}^k\|_H^2 = \frac{1}{\beta}\|\lambda^{k+1} - \tilde{\lambda}^k\|^2 = \frac{1}{\beta}\|\beta B(y^k - \tilde{y}^k)\|^2 = \beta\|B(y^k - \tilde{y}^k)\|^2.$$

On the other hand

$$\|v^k - \tilde{v}^k\|_H^2 = \beta\|B(y^k - \tilde{y}^k)\|^2 + \frac{1}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2.$$

Therefore, we have $\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2 = \frac{1}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2$ and (5.7) is true. The assertion of this lemma is proved. $\square$

Now, we are ready to present the main result regarding the convergence rate of the ADM (2.1), as the following theorem shows.

**Theorem 5.2** *Let $\{v^k\}$ be the sequence generated by the ADM (2.1) and the associated sequence $\{\tilde{w}^k\}$ be defined by (4.1). For any integer number $t > 0$, let $\tilde{w}_t$ be defined by*

$$\tilde{w}_t = \frac{1}{t+1} \sum_{k=0}^{t} \tilde{w}^k, \tag{5.8}$$

*where $\tilde{w}^k$ is defined in (4.1). Then, $\tilde{w}_t \in \Omega$ and*

$$\theta(\tilde{u}_t) - \theta(u) + (\tilde{w}_t - w)^T F(w) \le \frac{1}{2(t+1)} \|v - v^0\|_H^2, \quad \forall w \in \Omega, \tag{5.9}$$

where $H$ is given by (1.5).

**Proof**. First, because of (4.1) and $w^k \in \Omega$, it holds that $\tilde{w}^k \in \Omega$ for all $k \ge 0$. Thus, together with convexity of $\mathcal{X}$ and $\mathcal{Y}$, (5.8) implies that $\tilde{w}_t \in \Omega$. Second, the inequalities (5.4) and (5.5) imply that

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(w) + \frac{1}{2}\|v - v^k\|_H^2 \ge \frac{1}{2}\|v - v^{k+1}\|_H^2, \ \forall w \in \Omega. \tag{5.10}$$



Summing the inequality (5.10) over $k = 0, 1, \ldots, t$, we obtain

$$(t+1)\theta(u) - \sum_{k=0}^{t} \theta(\tilde{u}^k) + \Big((t+1)w - \sum_{k=0}^{t} \tilde{w}^k\Big)^T F(w) + \frac{1}{2}\|v - v^0\|_H^2 \ge 0,$$

for any $w \in \Omega$. Use the notation of $\tilde{w}_t$, it can be written as

$$\frac{1}{t+1} \sum_{k=0}^{t} \theta(\tilde{u}^k) - \theta(u) + (\tilde{w}_t - w)^T F(w) \le \frac{1}{2(t+1)} \|v - v^0\|_H^2, \ \forall w \in \Omega. \tag{5.11}$$

Since $\theta(u)$ is convex and

$$\tilde{u}_t = \frac{1}{t+1} \sum_{k=0}^{t} \tilde{u}^k,$$

we have that

$$\theta(\tilde{u}_t) \le \frac{1}{t+1} \sum_{k=0}^{t} \theta(\tilde{u}^k).$$

Substituting it in (5.11), the assertion of this theorem follows directly. $\square$

Recall (5.3). The conclusion (5.9) thus indicates obviously that the ADM (2.1) is able to generate an approximate solution (i.e., $\tilde{w}_t$) of (5.8) with the accuracy $O(1/t)$ after $t$

iterations. That is, the convergence rate $O(1/t)$ of the ADM (2.1) is established.

# 6 Convergence rate in the non-ergodic sense

This section shows that the sequence $\{\|v^k - v^{k+1}\|_H^2\}$ is monotonically non-increasing.

$$\|v^k - v^{k+1}\|_H^2 \le \|v^{k-1} - v^k\|_H^2, \quad \forall\, k \ge 1. \tag{6.1}$$

Based on (3.8) and (6.1), we drive

$$\|v^k - v^{k+1}\|_H^2 \le \frac{1}{(k+1)}\|v^0 - v^*\|_H^2, \quad \forall v^* \in \mathcal{V}^*.$$

Since $\|v^k - v^{k+1}\|_H^2$ is viewed as the stopping criterium, we obtain the worst-case $O(1/t)$ convergence rate in a non-ergodic sense.

Again, we prove several lemmas for this purpose. First of all, we observe that $v^{k+1}$ and $\tilde{v}^k$ defined in (4.1) are related by (as pointed in (4.3))

$$v^k - v^{k+1} = M(v^k - \tilde{v}^k), \tag{6.2}$$



where the matrix $M$ is given in (1.6).

Lemma 4.1 enables us to establish an important inequality in the following lemma.

**Lemma 6.1** *Let $\{v^k\}$ be the sequence generated by (2.1), the associated sequence $\{\tilde{w}^k\}$ be defined by (4.1) and $Q_0$ be given in (1.4). Then, we have*

$$(\tilde{w}^k - \tilde{w}^{k+1})^T Q_0 \{(v^k - v^{k+1}) - (\tilde{v}^k - \tilde{v}^{k+1})\} \ge 0. \tag{6.3}$$

**Proof**. Set $w = \tilde{w}^{k+1}$ in (4.4), we have

$$\theta(\tilde{u}^{k+1}) - \theta(\tilde{u}^k) + (\tilde{w}^{k+1} - \tilde{w}^k)^T \{F(\tilde{w}^k) + Q_0(\tilde{v}^k - v^k)\} \ge 0. \tag{6.4}$$

Note that (4.4) is also true for $k := k+1$ and thus we have

$$\theta(u) - \theta(\tilde{u}^{k+1}) + (w - \tilde{w}^{k+1})^T \{F(\tilde{w}^{k+1}) + Q_0(\tilde{v}^{k+1} - v^{k+1})\} \ge 0, \quad \forall\, w \in \Omega.$$

Set $w = \tilde{w}^k$ in the above inequality, we obtain

$$\theta(\tilde{u}^k) - \theta(\tilde{u}^{k+1}) + (\tilde{w}^k - \tilde{w}^{k+1})^T \{F(\tilde{w}^{k+1}) + Q_0(\tilde{v}^{k+1} - v^{k+1})\} \ge 0. \tag{6.5}$$

Adding (6.4) and (6.5) and using the monotonicity of $F$, we get (6.3) immediately. $\square$

**Lemma 6.2** *Let $\{v^k\}$ be the sequence generated by (2.1), the associated sequence $\{\tilde{w}^k\}$ be defined by (4.1), the matrices $H$, $Q$ and $M$ be given in (1.5) and (1.6). Then, we have*

$$\begin{aligned}
&(v^k - \tilde{v}^k)^T M^T H M \{(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\} \\
\geq\ & \frac{1}{2}\|(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\|^2_{(Q^T+Q)},
\end{aligned} \tag{6.6}$$

*where the matrix $Q$ is defined in* (1.5).

**Proof**. First, according to the structures of $Q_0$ and $Q$, it follows from (6.3) that

$$(\tilde{v}^k - \tilde{v}^{k+1})^T Q\{(v^k - v^{k+1}) - (\tilde{v}^k - \tilde{v}^{k+1})\} \geq 0. \tag{6.7}$$

Adding the term

$$\{(v^k - v^{k+1}) - (\tilde{v}^k - \tilde{v}^{k+1})\}^T Q\{(v^k - v^{k+1}) - (\tilde{v}^k - \tilde{v}^{k+1})\}$$

to the both sides of (6.7), and using $v^T Q v = \frac{1}{2} v^T (Q^T + Q) v$, we get

$$\begin{aligned}
&(v^k - v^{k+1})^T Q\{(v^k - v^{k+1}) - (\tilde{v}^k - \tilde{v}^{k+1})\} \\
\geq\ & \frac{1}{2}\|(v^k - v^{k+1}) - (\tilde{v}^k - \tilde{v}^{k+1})\|^2_{(Q^T+Q)}.
\end{aligned}$$



Reorder $(v^k - v^{k+1}) - (\tilde{v}^k - \tilde{v}^{k+1})$ to $(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})$, from the above inequality we get

$$\begin{aligned}
&(v^k - v^{k+1})^T Q\{(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\} \\
\geq\ & \frac{1}{2}\|(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\|^2_{(Q^T+Q)}.
\end{aligned}$$

Substituting the term $(v^k - v^{k+1})$ into the left-hand side of the last inequality, and using the relationship in (6.2) and the fact $Q = HM$ in (1.7), we obtain (6.6). $\square$

Finally, we are ready to show the assertion (6.1) in the following theorem.

**Theorem 6.1** *Let $\{v^k\}$ be the sequence generated by (2.1), the associated sequence $\{\tilde{w}^k\}$ be defined by (4.1). Then we have*

$$\|M(v^{k+1} - \tilde{v}^{k+1})\|^2_H \leq \|M(v^k - \tilde{v}^k)\|^2_H, \tag{6.8}$$

*and thus*

$$\|v^{k+1} - v^{k+2}\|^2_H \leq \|v^k - v^{k+1}\|^2_H. \tag{6.9}$$

**Proof**. Setting $a = M(v^k - \tilde{v}^k)$ and $b = M(v^{k+1} - \tilde{v}^{k+1})$ in the identity

$$\|a\|_H^2 - \|b\|_H^2 = 2a^T H(a-b) - \|a-b\|_H^2,$$

we obtain

$$\begin{aligned} &\|M(v^k - \tilde{v}^k)\|_H^2 - \|M(v^{k+1} - \tilde{v}^{k+1})\|_H^2 \\ &\quad = \quad 2(v^k - \tilde{v}^k)^T M^T H M\{(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\} \\ &\qquad\quad -\|M\{(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\}\|_H^2. \end{aligned}$$

Inserting (6.6) into the first term of the right-hand side of the last equality, we obtain

$$\begin{aligned} &\|M(v^k - \tilde{v}^k)\|_H^2 - \|M(v^{k+1} - \tilde{v}^{k+1})\|_H^2 \\ &\quad \geq \quad \|(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\|_{(Q^T+Q)}^2 \\ &\qquad\quad -\|M\{(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\}\|_H^2 \\ &\quad = \quad \|(v^k - \tilde{v}^k) - (v^{k+1} - \tilde{v}^{k+1})\|_{\{(Q^T+Q)-M^T H M\}}^2. \end{aligned}$$

Because of the positive definiteness of the matrix $(Q^T + Q) - M^T HM$ ( indicated in (1.8)). Recall the relationship in (6.2). The assertion (6.9) follows immediately from (6.8).
□



With Theorems 3.1 and 6.1, we can prove a worst-case $O(1/t)$ convergence rate in a non-ergodic sense for the ADM scheme (2.1).

**Theorem 6.2** *Let $\{v^k\}$ be the sequence generated by (2.1). Then we have*

$$\|v^k - v^{k+1}\|_H^2 \leq \frac{1}{(k+1)}\|v^0 - v^*\|_H^2, \quad \forall v^* \in \mathcal{V}^*. \tag{6.10}$$

**Proof**. First, it follows from (3.8) that

$$\sum_{t=0}^{\infty} \|v^t - v^{t+1}\|_H^2 \leq \|v^0 - v^*\|_H^2, \quad \forall\, v^* \in \mathcal{V}^*. \tag{6.11}$$

According to Theorem 6.1, the sequence $\{\|v^t - v^{t+1}\|_H^2\}$ is monotonically non-increasing. Therefore, we have

$$(k+1)\|v^k - v^{k+1}\|_H^2 \leq \sum_{i=0}^{k} \|v^i - v^{i+1}\|_H^2. \tag{6.12}$$

The assertion (6.10) follows from (6.11) and (6.12) immediately. □

Notice that $\mathcal{V}^*$ is convex and closed. Let $d := \inf\{\|v^0 - v^*\|_H \mid v^* \in \Omega^*\}$. Then, for any given $\epsilon > 0$, Theorem 6.2 shows that the ADM scheme (2.1) needs at most $\lfloor d^2/\epsilon \rfloor$ iterations to ensure that $\|v^k - v^{k+1}\|_H^2 \le \epsilon$. Recall that $v^{k+1}$ is a solution of VI$(\Omega, F, \theta)$ if $\|v^k - v^{k+1}\|_H^2 = 0$ (see Lemma 3.1). A worst-case $O(1/t)$ convergence rate in a non-ergodic sense for the ADM scheme (2.1) is thus established in Theorem 6.2.

# References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, manuscript, 2010.

[2] C.H. Chen, B.S. He, and X.M. Yuan, Matrix completion via alternating direction methods, IMA J. Numer. Anal., 32,. 227-245, 2012.

[3] J. Eckstein and D.P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Program., 55 (1992), pp. 293-318.

[4] E. Esser, *Applications of Lagrangian-based alternating direction methods and connections to split Bregman*, UCLA CAM Report, 2009.

[5] F. Facchinei and J. S. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, Vol. I. Springer Series in Operations Research, Springer Verlag, New York, 2003.



[6] M. Fortin and R. Glowinski, Augmented Lagrangian Methods: Applications to the Numerical Solutions of Boundary Value Problems, Stud. Math. Appl. 15, North – Holland, Amsterdam, 1983.

[7] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appli., 2 (1976), pp. 17-40.

[8] R. Glowinski, Numerical methods for Nonlinear Variational Problems, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[9] R. Glowinski and P. Le Tallec, Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics, SIAM Studies in Applied Mathematics, Philadelphia, PA, 1989.

[10] B. S. He, L. Z. Liao, D. R. Han and H. Yang, *A new inexact alternating directions method for monontone variational inequalities*, Math. Program., 92 (2002), pp. 103-118.

[11] B.S. He, M. Tao and X.M. Yuan, Alternating Direction Method with Gaussian Back Substitution for Separable Convex Programming, SIAM J. Optim. 22, 313-340, 2012

[12] B. S. He, M. H. Xu and X. M. Yuan, Solving large-scale least squares covariance matrix problems by alternating direction methods, SIAM J. Matrix Analy. Appli., 32, (2011), pp. 136-152.

[13] B.S. He and X.M. Yuan, On the $O(1/n)$ Convergence Rate of the Douglas-Rachford Alternating Direction Method， SIAM J. Numer. Anal. 50, 700-709, 2012.

[14] B.S. He and X.M. Yuan, Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective, SIAM J. Imag. Sci., 5, 119-149, 2012.

[15] B.S. He and X.M. Yuan, On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers http://www.optimization-online.org/DB_HTML/2012/01/3318.html

[16] M. R. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appli., 4(1969), pp. 303-320.

[17] S. Kontogiorgis and R. R. Meyer, A variable-penalty alternating directions method for convex optimization, Math. Program., 83 (1998), pp. 29-53.

[18] A. Nemirovski, *Prox-method with rate of convergence $O(1/t)$ for variational inequality with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems*, SIAM J. Optim. 15 (2005), pp. 229-251.

[19] Y. E. Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence O($1/k^2$)*, Doklady AN SSSR, 269 (1983), pp. 543-547.

[20] M. Tao and X.M. Yuan, *Recovering low-rank and sparse components of matrices from incomplete and noisy observations*, SIAM J. Optim., 21 (2011), pp. 57-81.

[21] P. Tseng, *On accelerated proximal gradient methods for convex-concave optimization*, manuscript, University of Washington, USA, 2008.

[22] X. M. Yuan, *Alternating direction methods for sparse covariance selection*, J. of Scientific Computing, **51**, 261-273, 2012.



**热忱希望读者指正！**