

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 fundamental 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 publications [4, 6]

1 投影与变分不等式的一些基本性质

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

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

的求解方法. 一个变分不等式称为单调的, 是指 $\text{VI}(\Omega, F)$ 中的 F 是 \mathfrak{R}^n (或 Ω) 上的单调算子 (monotone operator), 即 F 满足

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

说 F 是单调仿射算子, 指 $F(u) = Mu + q$, $q \in \mathfrak{R}^n$, 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 $u \in \mathfrak{R}^n$ 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

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

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

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

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

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

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 Ω 上的投影, 也就是说

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

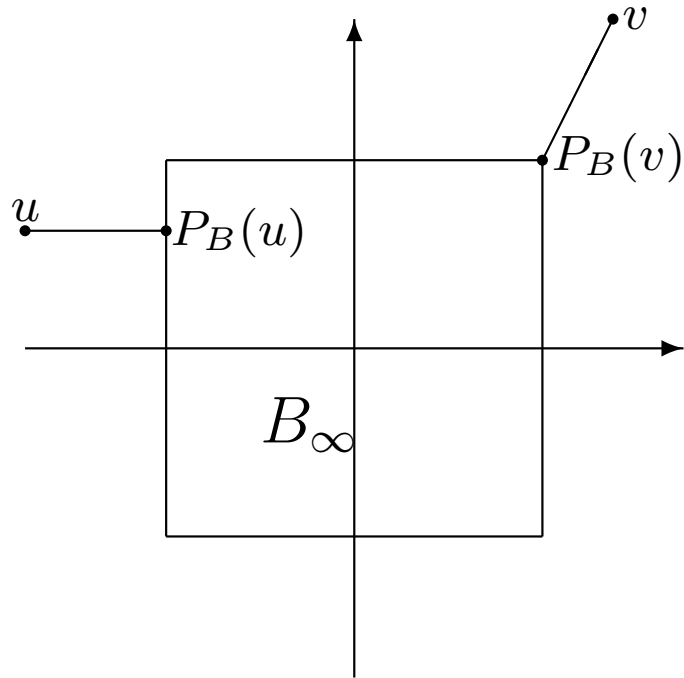
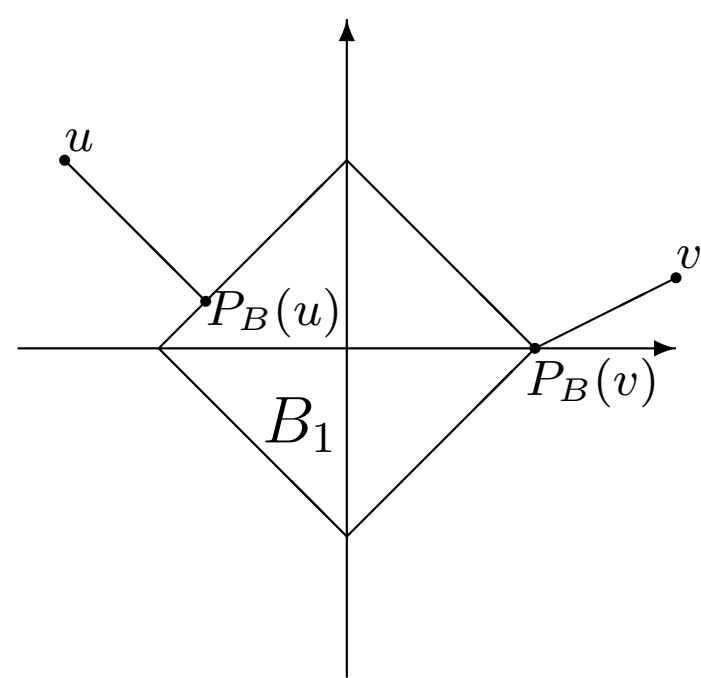
如果 $\Omega = \mathfrak{R}_+^n$ (n -维空间的非负卦限), 那么 $P_\Omega(v)$ 的每个分量为

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

如果 Ω 是 n -维空间中以 c 为球心半径为 r 的球, 那么

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

在 l_∞ 和 l_1 模意义下的“单位球”上投影如下图所示:

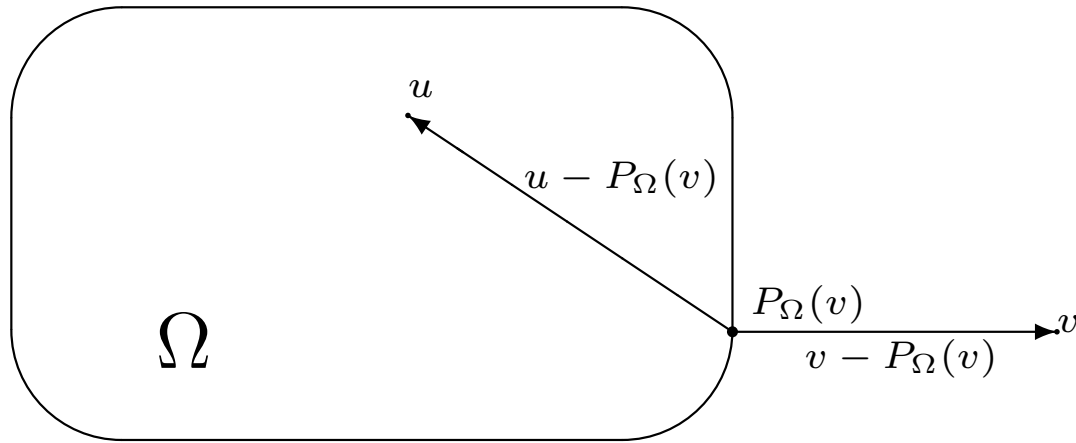
Fig. 1. Projection on B_∞ Fig. 2. Projection on B_1

通篇, 我们记

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

引理 1.1 设 $\Omega \subset \mathbb{R}^n$ 是闭凸集, 则有

$$(v - P_{\Omega}(v))^T (u - P_{\Omega}(v)) \leq 0 \quad \forall v \in \mathbb{R}^n, \forall u \in \Omega. \quad (1.5)$$



不等式(1.5)的几何解释.

证明. 首先, 根据 $P_{\Omega}(v)$ 的定义, 有

$$\|v - P_{\Omega}(v)\| \leq \|v - w\|, \quad \forall w \in \Omega. \quad (1.6)$$

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

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

对这个 w , 利用 (1.6), 就有

$$\|v - P_{\Omega}(v)\|^2 \leq \|v - P_{\Omega}(v) - \theta(u - P_{\Omega}(v))\|^2.$$

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

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

令 $\theta \rightarrow 0_+$, 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.

引理 1.2 设 $\Omega \subset \mathbb{R}^n$ 是闭凸集, 则有

$$\|P_{\Omega}(v) - P_{\Omega}(u)\| \leq \|v - u\|, \quad \forall u, v \in \mathbb{R}^n. \quad (1.7)$$

$$\|P_{\Omega}(v) - u\| \leq \|v - u\|, \quad \forall v \in \mathbb{R}^n, u \in \Omega. \quad (1.8)$$

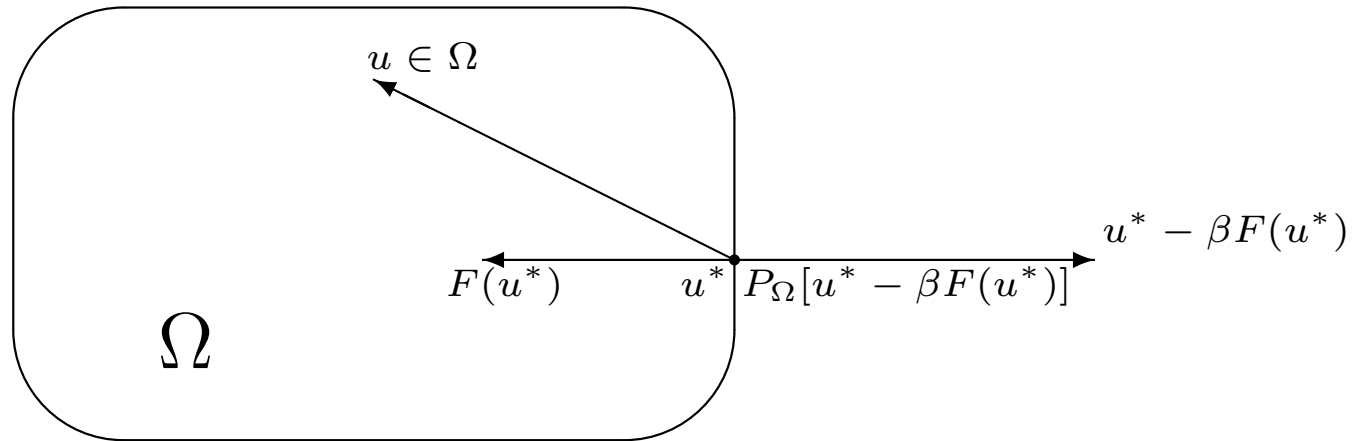
$$\|P_{\Omega}(v) - u\|^2 \leq \|v - u\|^2 - \|v - P_{\Omega}(v)\|^2, \quad \forall v \in \mathbb{R}^n, u \in \Omega. \quad (1.9)$$

我们把这几个性质留给读者自己去证明.

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 Ω^* 非空. 单调变分不等式的解集是凸的, 这个结论可以在 [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

$$u \in \Omega^* \Leftrightarrow u = P_{\Omega}[u - \beta F(u)].$$



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

$$e(u, \beta) := u - P_{\Omega}[u - \beta F(u)] \quad (1.10)$$

的一个零点 u^* , 后面我们会给出证明. 因此, 对确定的 $\beta > 0$, $\|e(u, \beta)\|$ 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 $e(u)$.

定理 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

$$(v - P_{\Omega}(v))^T (u^* - P_{\Omega}(v)) \leq 0, \quad \forall v \in R^n.$$

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

$$\|e(u^*, \beta)\|^2 \leq \beta e(u^*, \beta)^T F(u^*). \quad (1.11)$$

另一方面, 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

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

即

$$e(u^*, \beta)^T F(u^*) \leq 0. \quad (1.12)$$

由不等式 (1.11) 和 (1.12) 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

$$\{e(u^*, \beta) - \beta F(u^*)\}^T \{u - P_\Omega[u^* - \beta F(u^*)]\} \leq 0, \quad \forall u \in \Omega. \quad (1.13)$$

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 (1.13), 可以得到

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

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

下面的定理说明 $\|e(u, \beta)\|$ 是 β 的不减函数, 而 $\{\|e(u, \beta)\|/\beta\}$ 是 β 的不增函数. 这个简单证明只用到一元二次不等式的初等知识和工具不等式 (1.5), 它初见于 [14].

定理 1.2 对所有的 $u \in \mathfrak{R}^n$ 和 $\tilde{\beta} \geq \beta > 0$, 我们有

$$\|e(u, \tilde{\beta})\| \geq \|e(u, \beta)\| \quad (1.14)$$

和

$$\frac{\|e(u, \tilde{\beta})\|}{\tilde{\beta}} \leq \frac{\|e(u, \beta)\|}{\beta}. \quad (1.15)$$

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

$$(t - 1)(t - \tilde{\beta}/\beta) \leq 0 \quad (1.16)$$

的解. 首先, 由工具不等式 (1.5), 我们有

$$(v - P_{\Omega}(v))^T (P_{\Omega}(v) - w) \geq 0, \quad \forall w \in \Omega. \quad (1.17)$$

在 (1.17) 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

$$\{e(u, \beta) - \beta F(u)\}^T \{e(u, \tilde{\beta}) - e(u, \beta)\} \geq 0. \quad (1.18)$$

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

$$\{e(u, \tilde{\beta}) - \tilde{\beta}F(u)\}^T \{e(u, \beta) - e(u, \tilde{\beta})\} \geq 0. \quad (1.19)$$

分别将不等式 (1.18) 和 (1.19) 乘上 $\tilde{\beta}$ 和 β , 然后再将它们相加, 我们得到

$$\{\tilde{\beta}e(u, \beta) - \beta e(u, \tilde{\beta})\}^T \{e(u, \tilde{\beta}) - e(u, \beta)\} \geq 0 \quad (1.20)$$

并有

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

对上式采用 Cauchy-Schwarz 不等式, 就有

$$\beta \|e(x, \tilde{\beta})\|^2 - (\beta + \tilde{\beta}) \|e(x, \beta)\| \cdot \|e(x, \tilde{\beta})\| + \tilde{\beta} \|e(x, \beta)\|^2 \leq 0. \quad (1.21)$$

将 (1.21) 除 $\beta \|e(x, \beta)\|^2$, 并利用 t 的定义便得

$$t^2 - \left(1 + \frac{\tilde{\beta}}{\beta}\right)t + \frac{\tilde{\beta}}{\beta} \leq 0.$$

因此不等式 (1.16) 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

$$(F11) \quad (\tilde{u} - u^*)^T \beta F(u^*) \geq 0$$

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

$$(F12) \quad (\tilde{u} - u^*)^T ([u - \beta F(u)] - \tilde{u}) \geq 0$$

因为我们考虑的是单调变分不等式, 根据单调算子的性质, 有

$$(F13) \quad (\tilde{u} - u^*)^T (\beta F(\tilde{u}) - \beta F(u^*)) \geq 0$$

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 .

根据定理 1.1, 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

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

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

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

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

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

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

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

对任意给定的确定解点 u^* , (2.3) 表明 $\vartheta_k(\alpha)$ 是 α 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

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

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

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

使二次函数 $q_k(\alpha)$ 达到极大的 α_k^* 是

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

如果取

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

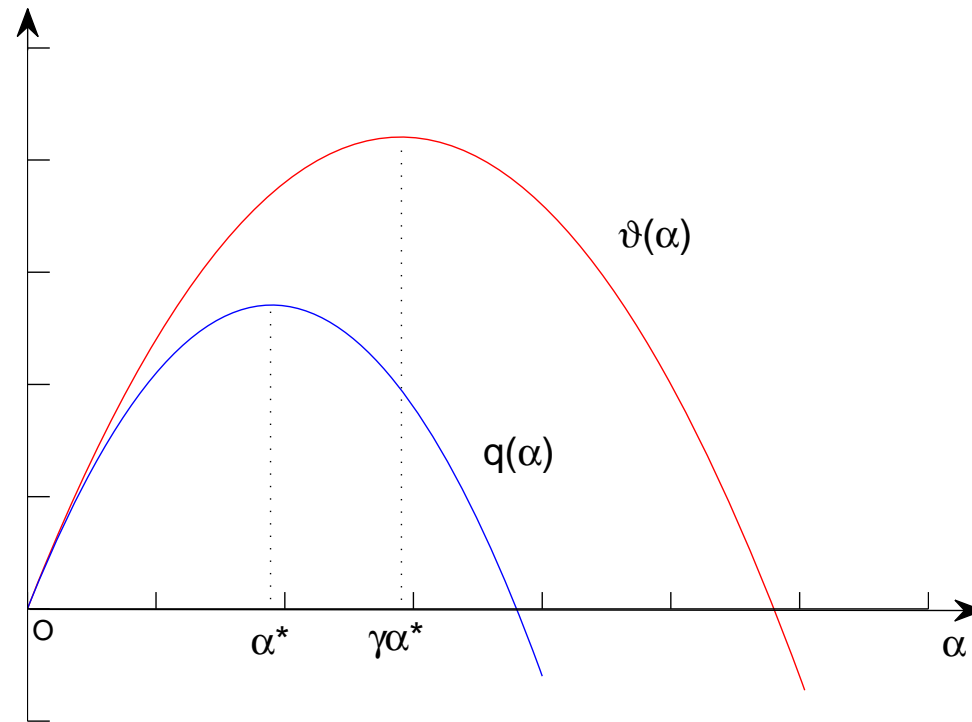
由 (2.5) 和 (2.6), 得到

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

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

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

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

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

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

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

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

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

初等收缩算法.

考虑将 (2.1a) 中的 $\varphi(u^k, \tilde{u}^k) \geq \delta\|u^k - \tilde{u}^k\|^2$ 改成条件

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

我们将条件 (2.1b) 和 (2.10) 满足时, 用单位步长的迭代公式

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

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

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

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.

定理 2.1 设 $VI(\Omega, F)$ 的解集 Ω^* 非空, 并且有 $\|d(u^k, \tilde{u}^k)\| = \mathcal{O}\|u^k - \tilde{u}^k\|$, 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

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

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,

$$\text{只要求 } M + M^T \text{ 半正定.}$$

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, 我们会在第十讲中讨论.

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

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

相加, 对所有的 $u \in \mathfrak{R}^n$, 都有

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

由上式直接得到

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

由于 M 半正定, 所以

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

令

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

和

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

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 α_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

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

在这个算法中, 参数 β 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 β_k , 使得

$$\beta_k \|M^T(u^k - \tilde{u}^k)\| = \mu \cdot \|u^k - \tilde{u}^k\|, \quad \mu \in [0.5, 2].$$

♣ 如果所取的 β 满足

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

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

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

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

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

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 F11 和 F12 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

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

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

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

这个 G -模下的收缩算法, 不用计算步长. 迭代式 (3.5) 相当于求解线性方程组

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

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

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

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

$$u^{k+1} = u^k - \gamma \alpha_k^* (u^k - \tilde{u}^k), \quad \alpha_k^* = \frac{\|u^k - \tilde{u}^k\|^2}{\|u^k - \tilde{u}^k\|_G^2}, \quad \gamma \in (0, 2) \quad (3.7)$$

产生新的迭代点. 利用 (3.6) 就有

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

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

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

上式是证明方法收敛的关键不等式.

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

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

相加, 对所有的 $u \in \mathfrak{R}^n$, 都有

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

我们定义

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

和

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

从而对一切的 $u \in \mathfrak{R}^n$, 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

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

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

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

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

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

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

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

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, 12]. 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 [15]. 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 [12]. 感谢 Sun 当年在论文 [12] 的 Footnote 和参考文献中都提到 [6].

大道至简!

The construction of projection contraction algorithms is based on three fundamental inequalities. Its principle is simple and unified, it gives us beautiful enjoyment !

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 (宾夕法尼亚大学, 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:

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

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

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

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

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

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

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

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

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

and thus

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

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

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

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

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

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

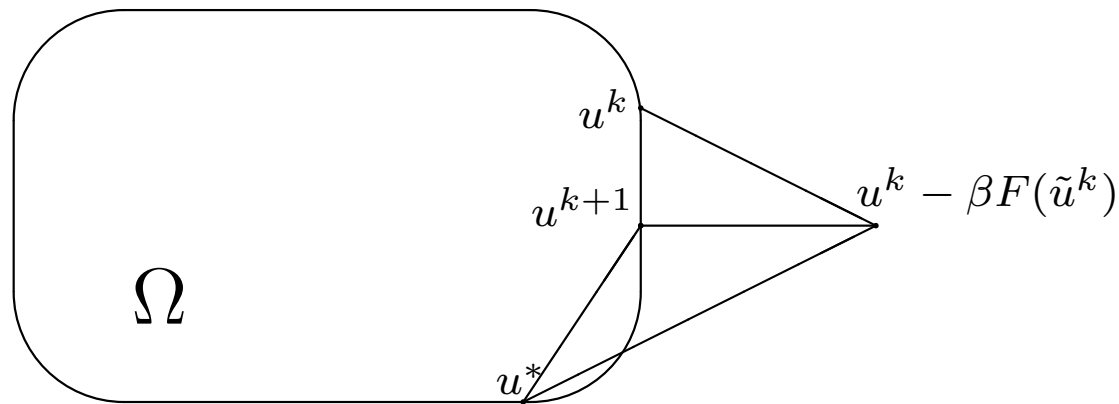
$$\beta \|F(u^k) - F(\tilde{u}^k)\| \leq \nu \|u^k - \tilde{u}^k\|, \quad \nu \in (0, 1). \quad (5.6)$$

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 (1.9).

♣ 首先, 由于 u^{k+1} 是 $u^k - \beta F(\tilde{u}^k)$ 到 Ω 上的投影, 根据投影性质 (1.9) 有

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



Geometric interpretation of inequality (5.7)

化简 (5.7) 式, 得到

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

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

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

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

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

♣ 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 Ω 上的投影并且 $u^{k+1} \in \Omega$, 根据 (1.5), 有

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

将上式加到 (5.9) 的右端, 得到

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

♣ 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 (1.9). 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 $\Omega = \mathfrak{R}_+^n$ 的特殊形式）

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

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 $D(u)$ 分别是 $F(u)$ 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 [13]^b.

^aIn the paper by Harker and Pang [2], the matrix $M = A^T A + B + D$, 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 $d_{jj} \equiv 0$.

^bIn [13], the components of nonlinear mapping $D(u)$ are $D_j(u) = \text{constat} * \arctan(u_j)$. 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 $r_k \leq \mu$ **then** $\beta_k := \beta_k * 1.5,$ **end(if)**

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

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 **If** $r_k \leq \mu$ **then** $\beta_k := \beta_k * 1.5$ **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 $r_k > \nu$, $\beta_k := \frac{2}{3}\beta_k * \min\{1, \frac{1}{r_k}\}$,

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

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

end(while)

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

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

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

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.

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))

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

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))

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

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('Extragradient 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'*dx; r2=dx'*dx; 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] J. Stoer and C. Witzgall, *Convexity and Optimization in Finite Dimension I*. Springer-Verlag, 1970.
- [11] D.F. Sun, A new step-size skill for solving a class of nonlinear projection equations, JCM, **13**, 357-368, 1995.
- [12] D.F. Sun, A class of iterative methods for solving nonlinear projection equations, JOTA **91**, 123-140, 1996.
- [13] K. Taji, M. Fukushima, and T. Ibaraki, A globally convergent Newton method for solving strongly monotone variational inequalities, Math. Progr. **58**, 369-383, 1993.
- [14] T. Zhu and Z. G. Yu, A simple proof for some important properties of the projection mapping. Math. Inequal. Appl. **7**, 453–456, 2004.
- [15] 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.