

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

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

与 F 经调比后的问题

$$\text{VI}(\Omega, F) \quad u^* \in \Omega, \quad (u - u^*)^T \beta F(u^*) \geq 0, \quad \forall u \in \Omega \quad (0.2)$$

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, 17]. 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 [18, 19]. 本文分别介绍单调线性变分不等式和非线性变分不等式的两对孪生方法. 每对孪生方法都由投影提供一对孪生下降方向, 采用不同校正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, 从数学本身来讲, 有如此巧妙的内在联系, 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.

引理 1.1 设 $\Omega \subset R^n$ 是闭凸集, 则对任意的 $v \in \mathfrak{R}^n$, 有

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

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

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

求解单调变分不等式 (0.2) 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

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

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. 在 (1.1) 中令 $v = u^k - \beta_k F(u^k)$, 根据 $\tilde{u}^k = P_\Omega(v)$ 和前一讲说到的投影的性质, 我们有

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

进而得到

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

三个基本不等式 设 u^* 是变分不等式 VI(Ω, 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

$$(F1) \quad (\tilde{u}^k - u^*)^T \beta_k F(u^*) \geq 0. \quad (1.5)$$

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

$$(F2) \quad (\tilde{u}^k - u^*)^T ([u^k - \tilde{u}^k] - \beta_k F(u^k)) \geq 0. \quad (1.6)$$

根据单调算子的性质, 有

$$(F3) \quad (\tilde{u}^k - u^*)^T (\beta_k F(\tilde{u}^k) - \beta_k F(u^*)) \geq 0. \quad (1.7)$$

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:

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

此时所说的单调性是要求矩阵 $M^T + M$ 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 β ,

- **基于 F11 的上升方向**. 因为 $F(u) = Mu + q$, 基本不等式 (1.5) 可以写成

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

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

$$(u^k - u^*)^T \underline{\beta[M^T(u^k - \tilde{u}^k) + (Mu^k + q)]} \geq (u^k - \tilde{u}^k)^T \beta(Mu^k + q), \quad (2.1)$$

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

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

因此, 对 $u^k \in \Omega$, $\beta_k(M^T(u^k - \tilde{u}^k) + (Mu^k + q))$ 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 (F1+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

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

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

$$(u^k - u^*)^T \underline{(I + \beta M^T)}(u^k - \tilde{u}^k) \geq \|u^k - \tilde{u}^k\|^2. \quad (2.2)$$

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.

孪生方向 对线性变分不等式, (1.4) 可以写成

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

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

$$\begin{aligned} \tilde{u}^k \in \Omega, \quad (u - \tilde{u}^k)^T \beta [M^T (u^k - \tilde{u}^k) + (Mu^k + q)] \\ \geq (u - \tilde{u}^k)^T \underline{(I + \beta M^T)(u^k - \tilde{u}^k)}, \quad \forall u \in \Omega, \end{aligned} \quad (2.3)$$

我们称分处 (2.3) 两端的

$$\beta [M^T (u^k - \tilde{u}^k) + (Mu^k + q)] \quad \text{和} \quad (I + \beta M^T)(u^k - \tilde{u}^k) \quad (2.4)$$

为一对孪生方向. 它们分别是由 (F11) 和 (F11+F12) 产生的 (分别见 (2.1) 和 (2.2)). 为了方便, 我们记

$$g(u^k, \tilde{u}^k) = M^T (u^k - \tilde{u}^k) + (Mu^k + q). \quad (2.5)$$

利用这个表达式, (2.3)

$$\tilde{u}^k \in \Omega, \quad (u - \tilde{u}^k)^T \underline{\beta g(u^k, \tilde{u}^k)} \geq (u - \tilde{u}^k)^T \underline{(I + \beta M^T)(u^k - \tilde{u}^k)}, \quad \forall u \in \Omega. \quad (2.6)$$

2.2 利用 (F11+F12) 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 (F11+F12) 提供的方向产生新迭代点的公式为

$$u_I^{k+1}(\alpha) = u^k - \alpha(I + \beta M^T)(u^k - \tilde{u}^k). \quad (2.7)$$

为讨论步长 α 如何取, 我们将 (2.7) 产生的依赖于步长 α 的新迭代点记为 $u_I^{k+1}(\alpha)$. 考察与 α 相关的距离平方缩短量,

$$\vartheta_k(\alpha) := \|u^k - u^*\|^2 - \|u_I^{k+1}(\alpha) - u^*\|^2. \quad (2.8)$$

根据定义

$$\begin{aligned} \vartheta_k(\alpha) &= \|u^k - u^*\|^2 - \|u^k - u^* - \alpha(I + \beta M^T)(u^k - \tilde{u}^k)\|^2 \\ &= 2\alpha(u^k - u^*)^T(I + \beta M^T)(u^k - \tilde{u}^k) \\ &\quad - \alpha^2\|(I + \beta M^T)(u^k - \tilde{u}^k)\|^2. \end{aligned} \quad (2.9)$$

对任意给定的确定解点 u^* , (2.9) 表明 $\vartheta_k(\alpha)$ 是 α 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 (2.1), 我们有

定理 2.1 设 $u^{k+1}(\alpha)$ 由 (2.7) 生成. 对任意的 $\alpha > 0$, 由 (2.8) 定义的 $\vartheta_k(\alpha)$ 有

$$\vartheta_k(\alpha) \geq q_k^L(\alpha), \quad (2.10)$$

其中

$$q_k^L(\alpha) = 2\alpha\|u^k - \tilde{u}^k\|^2 - \alpha^2\|(I + \beta M^T)(u^k - \tilde{u}^k)\|^2. \quad (2.11)$$

证明. 这个结论可以从 (2.9) 利用 (2.1) 直接得到. \square

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 α_k^* 是

$$\alpha_k^* = \operatorname{argmax}\{q_k^L(\alpha)\} = \frac{\|u^k - \tilde{u}^k\|^2}{\|(I + \beta M^T)(u^k - \tilde{u}^k)\|^2}. \quad (2.12)$$

根据上式, 我们有

$$\alpha_k^* \geq \frac{1}{\|I + \beta M^T\|^2}. \quad (2.13)$$

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

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

$$u^{k+1} = u^k - \gamma \alpha_k^* (I + \beta M^T)(u^k - \tilde{u}^k), \quad (2.14)$$

下面给出方法收敛的关键性定理.

定理 2.2 设 u^{k+1} 由 (2.14) 生成, 其中步长 α_k^* 由 (2.12) 给出, 则有

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \frac{\gamma(2-\gamma)}{\|I + \beta M^T\|^2} \|u^k - \tilde{u}^k\|^2. \quad (2.15)$$

证明. 根据 (2.8) 和 (2.10), 由 (2.14) 产生的 u^{k+1} 满足

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - q_k^L(\gamma \alpha_k^*). \quad (2.16)$$

由 $q_k^L(\alpha)$ 和 α_k^* 的定义 (分别见 (2.11) 和 (2.12)), 得到

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

最后一个不等式用到了 (2.13). 定理得证. \square

2.3 利用 (F1) 提供的方向产生新的迭代点

§2.2 中的校正公式 (2.7) 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 (2.4))

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

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

$$u_{II}^{k+1}(\alpha) = P_{\Omega}\{u^k - \alpha\beta[M^T(u^k - \tilde{u}^k) + (Mu^k + q)]\}, \quad (2.17)$$

产生依赖于步长 α 的新的迭代点, 也保证属于 Ω . 为了区别与由 (2.7) 生成的新迭代点, 我们在 (2.17) 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

$$\zeta_k(\alpha) = \|u^k - u^*\|^2 - \|u_{II}^{k+1}(\alpha) - u^*\|^2 \quad (2.18)$$

看成是本次迭代的进步量, 它是步长 α 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 α , $\zeta_k(\alpha)$ ‘优于’ (2.10) 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.

定理 2.3 设 $u^{k+1}(\alpha)$ 由 (2.17) 生成. 对任意的 $\alpha > 0$, 由 (3.19) 定义的 $\zeta_k(\alpha)$ 有

$$\zeta_k(\alpha) \geq q_k^L(\alpha) + \|u_{II}^{k+1}(\alpha) - u_I^{k+1}(\alpha)\|^2, \quad (2.19)$$

其中 $q_k^L(\alpha)$, $u_I^{k+1}(\alpha)$, $u_{II}^{k+1}(\alpha)$ 分别由 (2.11), (2.7) 和 (2.17) 给出.

证明. 利用 (2.5) 的记号, (2.17) 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

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

然而, 根据投影的性质和余弦定理, 我们用更精细的关系式

$$\begin{aligned} \|u_{II}^{k+1}(\alpha) - u^*\|^2 &\leq \|u^k - \alpha\beta g(u^k, \tilde{u}^k) - u^*\|^2 \\ &\quad - \|u_{II}^{k+1}(\alpha) - (u^k - \alpha\beta g(u^k, \tilde{u}^k))\|^2. \end{aligned} \quad (2.20)$$

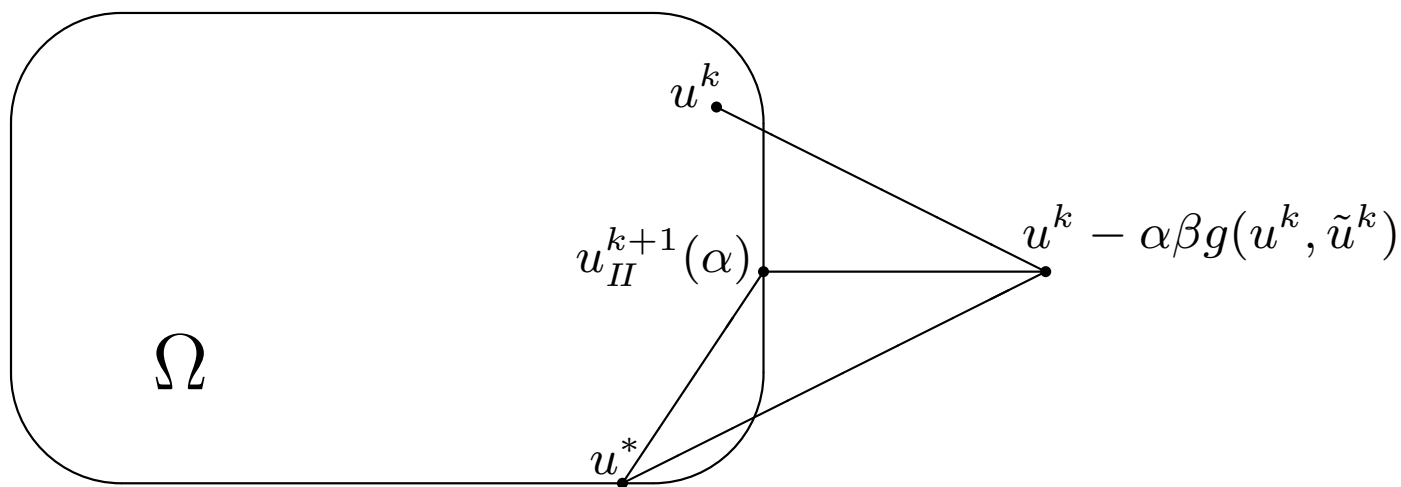


图 2. 不等式 (2.20) 的几何解释

因此, 利用 $\zeta_k(\alpha)$ 的定义(见 (2.18))和 (2.20), 我们有

$$\begin{aligned}
 \zeta_k(\alpha) &\geq \|u^k - u^*\|^2 - \|(u^k - u^*) - \alpha\beta g(u^k, \tilde{u}^k)\|^2 \\
 &\quad + \|(u_{II}^{k+1}(\alpha) - u^k) + \alpha\beta g(u^k, \tilde{u}^k)\|^2 \\
 &= 2\alpha\beta(u^k - u^*)^T g(u^k, \tilde{u}^k) + 2\alpha\beta(u_{II}^{k+1}(\alpha) - u^k)^T g(u^k, \tilde{u}^k) \\
 &\quad + \|u_{II}^{k+1}(\alpha) - u^k\|^2 \\
 &= \|u_{II}^{k+1}(\alpha) - u^k\|^2 + 2\alpha(u_{II}^{k+1}(\alpha) - u^*)^T \beta g(u^k, \tilde{u}^k). \quad (2.21)
 \end{aligned}$$

将 (2.21) 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

$$\begin{aligned} & (u_{II}^{k+1}(\alpha) - u^*)^T \beta g(u^k, \tilde{u}^k) \\ &= (u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T \beta g(u^k, \tilde{u}^k) + (\tilde{u}^k - u^*)^T \beta g(u^k, \tilde{u}^k). \end{aligned} \quad (2.22)$$

对于 (2.22) 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 (2.3), 有

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

也就是说,

$$\begin{aligned} (u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T \beta g(u^k, \tilde{u}^k) &\geq (u_{II}^{k+1}(\alpha) - u^k)^T (I + \beta M^T)(u^k - \tilde{u}^k) \\ &\quad + (u^k - \tilde{u}^k)^T (I + \beta M^T)(u^k - \tilde{u}^k). \end{aligned} \quad (2.23)$$

对于 (2.22) 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 (2.1), 即

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

和记号 $g(u^k, \tilde{u}^k)$, 我们得到

$$\begin{aligned}
& (\tilde{u}^k - u^*)^T \beta g(u^k, \tilde{u}^k) \\
& \geq (\tilde{u}^k - u^k)^T \beta \{M^T(u^k - \tilde{u}^k) + (Mu^k + q)\} + (u^k - \tilde{u}^k)^T \beta (Mu^k + q) \\
& \geq -\beta(u^k - \tilde{u}^k)^T M^T(u^k - \tilde{u}^k). \tag{2.24}
\end{aligned}$$

将 (2.23) 和 (2.24) 相加, 就有

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

将上面的不等式代入 (2.21), 利用 $q_k^L(\alpha)$ 的记号, 得到

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

再利用 (2.7), 就完成了定理结论 (2.19) 的证明. \square

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

定理 2.3 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

$$(PC-I) \quad u_I^{k+1} = u^k - \gamma\alpha_k^*(I + \beta M^T)(u^k - \tilde{u}^k) \quad (2.25)$$

或者

$$(PC-II) \quad u_{II}^{k+1} = P_\Omega [u^k - \gamma\alpha_k^*\beta[M^T(u^k - \tilde{u}^k) + (Mu^k + q)]] \quad (2.26)$$

产生新的迭代点 u^{k+1} , 其中的 α_k^* 都由 (2.12) 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 (2.25), 它的好处是生成 u^{k+1} 不用再做投影. 实际问题中, 到 Ω 上的投影代价往往不高 (例如 Ω 常常是一个正卦限或者框形), 因此常采用校正公式 (2.26). 这方面的理由我们在论文 [14] 中有更详细的说明.

根据 (2.3) 提供的孪生方向, 在采用相同步长的两个方法 (2.25) 和 (2.26) 中, 用记号 (2.7), (2.10) 和 (2.19) 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

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

其中 $q_k^L(\alpha)$ 由 (2.11) 给出. 对同样的 α , $\zeta_k(\alpha)$ 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, 17]. 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**.

计算实践 [5, 17] 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 β_k 选的能使

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

后面, 为了行文简单, 我们将 β_k 记成 β , 并假设这个 β 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 (F1+F3) 的上升方向**. 将基本不等式 (1.5) 和(1.6) 相加得到

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

从上式得到

$$(u^k - u^*)^T \beta F(\tilde{u}^k) \geq (u^k - \tilde{u}^k)^T \beta F(\tilde{u}^k). \quad (3.2)$$

如果 $u^k \in \Omega$, 根据 (1.4) 有

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

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

$$\begin{aligned} (u^k - \tilde{u}^k)^T \beta F(\tilde{u}^k) &= (u^k - \tilde{u}^k)^T \beta F(u^k) - (u^k - \tilde{u}^k)^T \beta (F(u^k) - F(\tilde{u}^k)) \\ &\geq (1 - \nu) \|u^k - \tilde{u}^k\|^2. \end{aligned} \quad (3.3)$$

不等式 (3.2) 和 (3.3) 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.

- **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

$$d(u^k, \tilde{u}^k) = (u^k - \tilde{u}^k) - \beta[F(u^k) - F(\tilde{u}^k)]. \quad (3.4)$$

随后得到

$$(u^k - u^*)^T d(u^k, \tilde{u}^k) \geq (u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k). \quad (3.5)$$

由 $d(u^k, \tilde{u}^k)$ 的表达式 (3.4) 和假设 (3.1), 根据 Cauchy-Schwarz 不等式, 有

$$(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k) \geq (1 - \nu) \|u^k - \tilde{u}^k\|^2. \quad (3.6)$$

不等式 (3.5) 和 (3.6) 告诉我们, 在条件 (3.1) 满足的情况下, 由 (3.4) 定义的 $d(u^k, \tilde{u}^k)$ 是欧氏模下的一个上升方向.

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

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

我们称分处 (3.7) 两端的

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

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

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

产生新的迭代点, 其中 $d(u^k, \tilde{u}^k)$ 是由 (3.4) 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 α . 考察与 α 相关的距离平方缩短量,

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

根据定义

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

对任意给定的确定解点 u^* , (3.10) 表明 $\vartheta_k(\alpha)$ 是 α 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.

定理 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

$$\vartheta_k(\alpha) \geq q_k^N(\alpha), \quad (3.11)$$

其中 $d(u^k, \tilde{u}^k)$ 由 (3.4) 给出,

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

证明. 这个结论可以从 (3.10) 利用 (3.5) 直接得到. \square

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 α_k^* 是

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

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

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

定理 3.2 设 u^{k+1} 由 (3.14) 生成. 对任意的 $\gamma \in (0, 2)$, 我们有

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

证明. 根据 (3.9) 和 (3.11), 由 (3.14) 产生的 u^{k+1} 满足

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - q_k^N(\gamma\alpha_k^*). \quad (3.15)$$

由 $q_k^N(\alpha)$ 和 α_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 (3.4) 和 (3.1), 可以得到

$$\begin{aligned} 2(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k) - \|d(u^k, \tilde{u}^k)\|^2 &= d(u^k, \tilde{u}^k)^T \{2(u^k - \tilde{u}^k) - d(u^k, \tilde{u}^k)\} \\ &= \{(u^k - \tilde{u}^k) - \beta[F(u^k) - F(\tilde{u}^k)]\}^T \{(u^k - \tilde{u}^k) + \beta[F(u^k) - F(\tilde{u}^k)]\} \\ &= \|u^k - \tilde{u}^k\|^2 - \beta_k^2 \| [F(u^k) - F(\tilde{u}^k)] \|^2 \geq (1 - \nu^2) \|u^k - \tilde{u}^k\|^2. \end{aligned} \quad (3.16)$$

因此, $\alpha_k^* > \frac{1}{2}$. 因此,

$$q_k^N(\gamma\alpha_k^*) > \frac{1}{2}\gamma(2 - \gamma)(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k).$$

由校正公式 (3.14) 产生的 u^{k+1} 满足

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \frac{1}{2}\gamma(2 - \gamma)(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k). \quad (3.17)$$

对上式中右端中的 $(u^k - \tilde{u}^k)^T d(u^k, \tilde{u}^k)$ 利用 (3.6), 就得到定理的结论. \square

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

$$u_{II}^{k+1}(\alpha) = P_{\Omega}[u^k - \alpha\beta F(\tilde{u}^k)], \quad (3.18)$$

为依赖于步长 α 的新的迭代点. 为了区别与由 (3.8) 生成的 u^{k+1} , 我们在 (3.18) 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

$$\zeta_k(\alpha) = \|u^k - u^*\|^2 - \|u_{II}^{k+1}(\alpha) - u^*\|^2 \quad (3.19)$$

看成是本次迭代的进步量, 它是步长 α 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 α , $\zeta_k(\alpha)$ ‘优于’ (3.11) 中的 $\vartheta_k(\alpha)$.

Theorem 3.1 设 $u^{k+1}(\alpha)$ 由 (3.18) 生成. 对任意的 $\alpha > 0$, 由 (3.19) 定义的 $\zeta_k(\alpha)$ 有

$$\zeta_k(\alpha) \geq q_k^N(\alpha) + \|u_{II}^{k+1}(\alpha) - u_I^{k+1}(\alpha)\|^2, \quad (3.20)$$

其中 $q_k^N(\alpha)$, $u_I^{k+1}(\alpha)$, $u_{II}^{k+1}(\alpha)$ 分别由 (3.12), (3.8) 和 (3.18) 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

$$\|u^{k+1}(\alpha) - u^*\|^2 \leq \|u^k - \alpha\beta F(\tilde{u}^k) - u^*\|^2 - \|u^k - \alpha\beta F(\tilde{u}^k) - u_{II}^{k+1}(\alpha)\|^2. \quad (3.21)$$

因此, 利用 $\zeta_k(\alpha)$ 的定义(见 (3.19)), 我们有

$$\begin{aligned} \zeta_k(\alpha) &\geq \|u^k - u^*\|^2 - \|(u^k - u^*) - \alpha\beta F(\tilde{u}^k)\|^2 + \|(u^k - u_{II}^{k+1}(\alpha)) - \alpha\beta F(\tilde{u}^k)\|^2 \\ &= 2\alpha(u^k - u^*)^T \beta F(\tilde{u}^k) + 2\alpha(u_{II}^{k+1}(\alpha) - u^k)^T \beta F(\tilde{u}^k) + \|u^k - u_{II}^{k+1}(\alpha)\|^2 \\ &= \|u^k - u_{II}^{k+1}(\alpha)\|^2 + 2\alpha(u_{II}^{k+1}(\alpha) - u^*)^T \beta F(\tilde{u}^k). \end{aligned} \quad (3.22)$$

将 (3.22) 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 (3.22) 的右端, 进一步得到

$$\zeta_k(\alpha) = \|u_{II}^{k+1}(\alpha) - u^k\|^2 + 2\alpha(u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T \beta F(\tilde{u}^k). \quad (3.23)$$

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

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

将它代入 (3.23) 的右端, 就有

$$\zeta_k(\alpha) \geq \|u_{II}^{k+1}(\alpha) - u^k\|^2 + 2\alpha(u_{II}^{k+1}(\alpha) - \tilde{u}^k)^T d(u^k, \tilde{u}^k). \quad (3.24)$$

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 (3.20) 的证明. \square

定理 3.1 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

$$(\text{PC Method-I}) \quad u_I^{k+1} = u^k - \gamma \alpha_k^* d(u^k, \tilde{u}^k) \quad (3.25)$$

或者

$$(\text{PC Method-II}) \quad u_{II}^{k+1} = P_{\Omega}[u^k - \gamma\alpha_k^*\beta_k F(\tilde{u}^k)] \quad (3.26)$$

产生新的迭代点 u^{k+1} , 其中的 α_k^* 都由 (3.13) 给出. 采用校正公式 (3.25), 它的好处是生成 u^{k+1} 不用再做投影. 实际问题中, 到 Ω 上的投影代价往往不高 (例如 Ω 常常是一个正卦限或者框形), 因此常采用校正公式 (3.26).

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

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

据此就可以证明下面的收敛性定理.

Theorem 3.2 设 $VI(\Omega, F)$ 的解集 Ω^* 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 [18, 19], 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 $b_{[i]}, i = 1, \dots, 10$ 是正则点 (regular points). 这些正则点的坐标是给定的. 点 $x_{[j]}, j = 1, \dots, 8$ 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 $x_{[j]}, j = 1, \dots, 8$ 的位置.

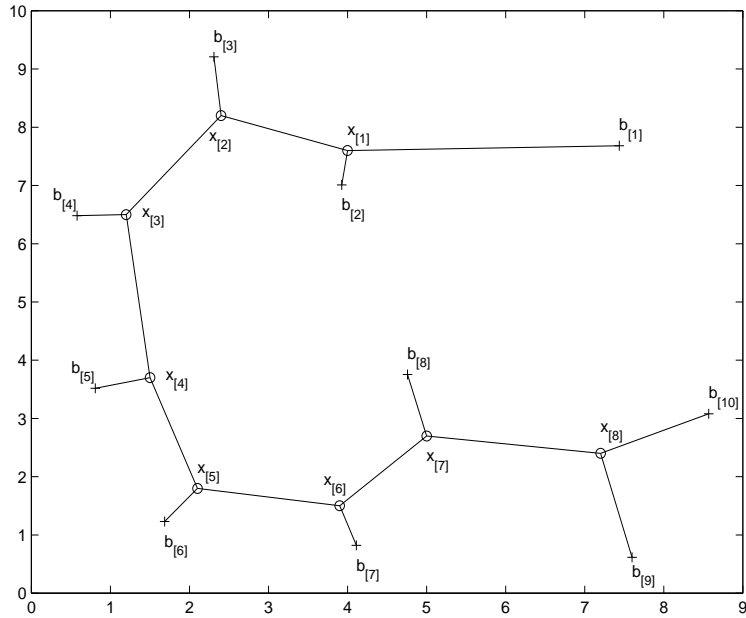


Fig. 1. Fig 1. Ordering of the topology

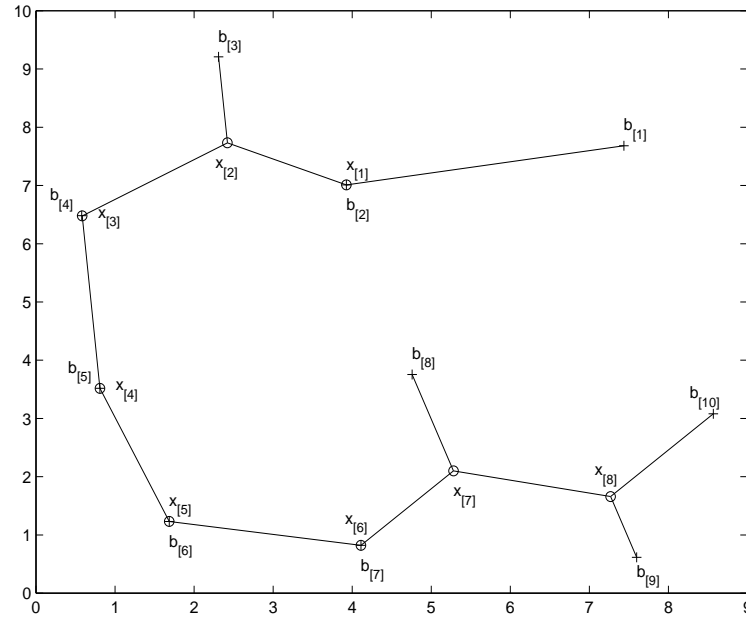


Fig. 2. Optimal solution in Euclidean-norm

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 (3.25) and II (3.26) are

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

and

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

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.

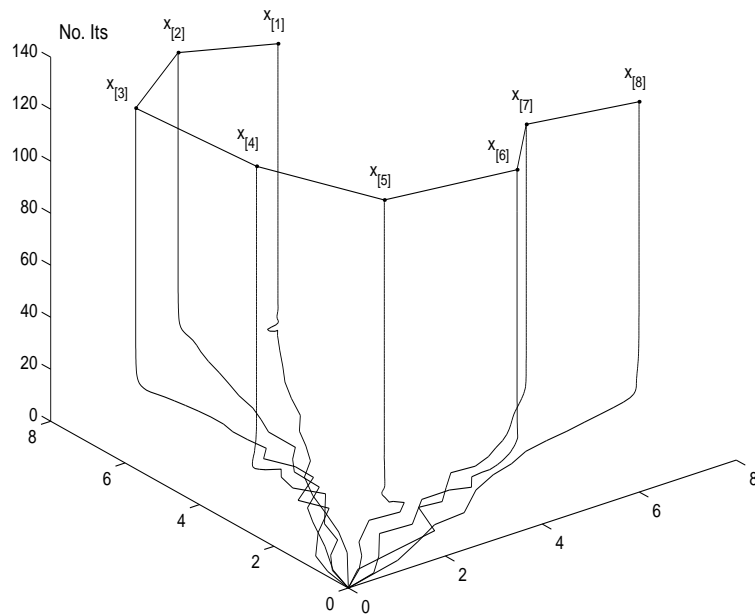


Fig. 3. Convergence tendency, $x^0 = 0$

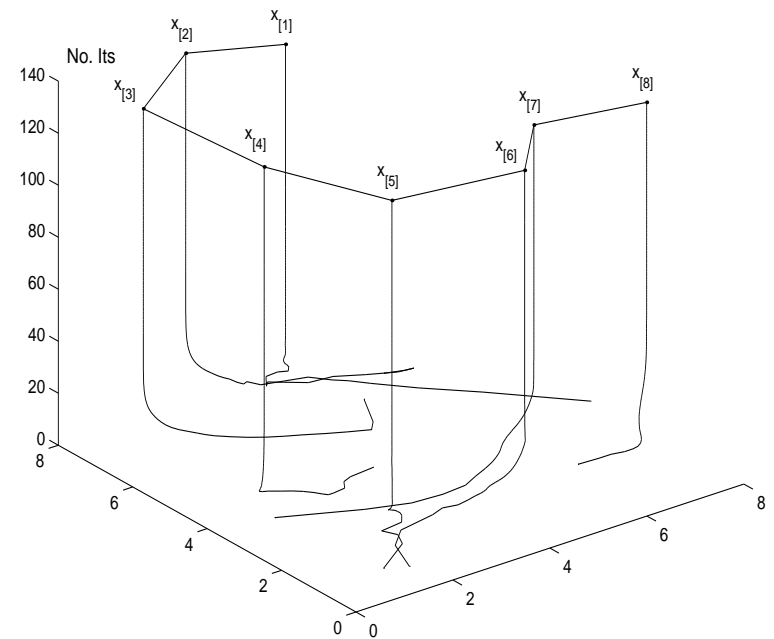


Fig. 4. Convergence tendency, x^0 random

对 l_1 -模和 l_∞ -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_∞ 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_∞ -norm, respectively.

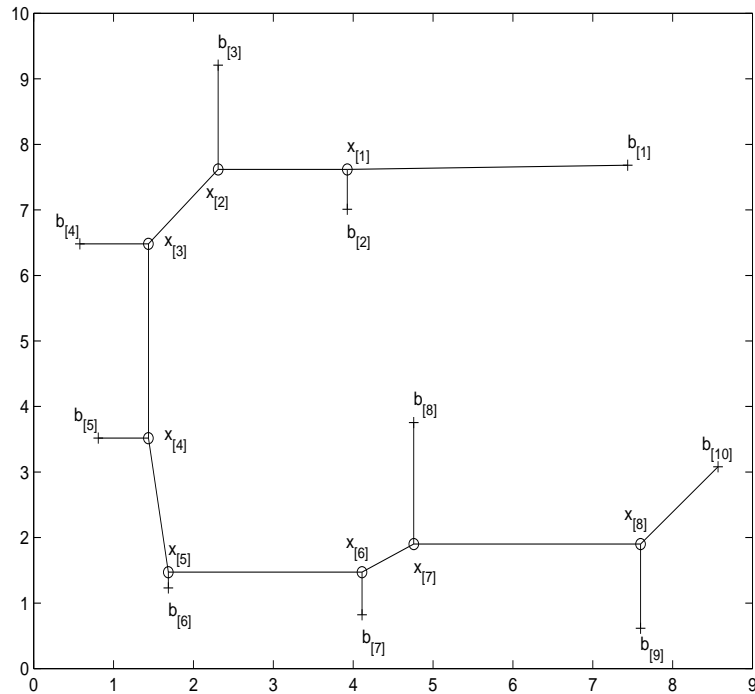


Fig. 5. Optimal solution, l_1 -norm

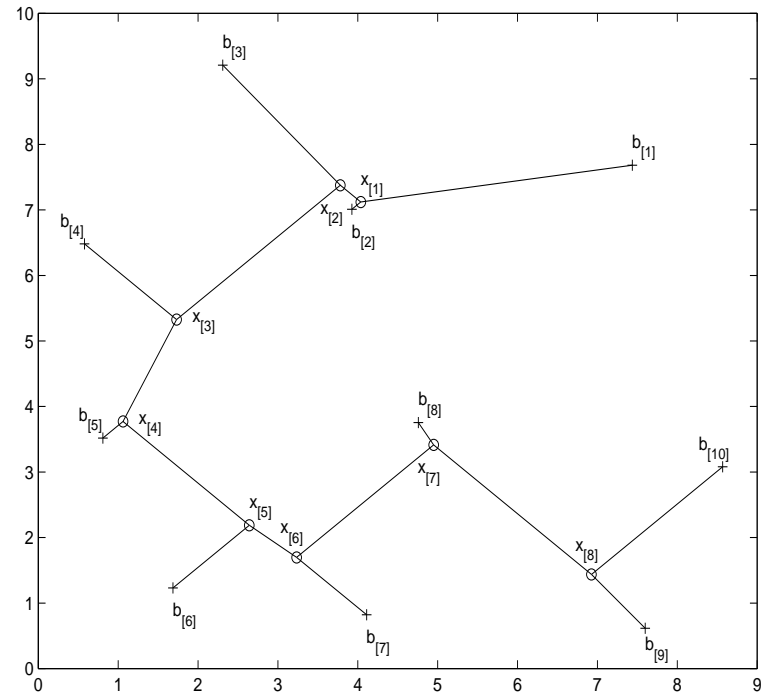


Fig. 6. Optimal solution, l_∞ -norm

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_∞ -norm, respectively.

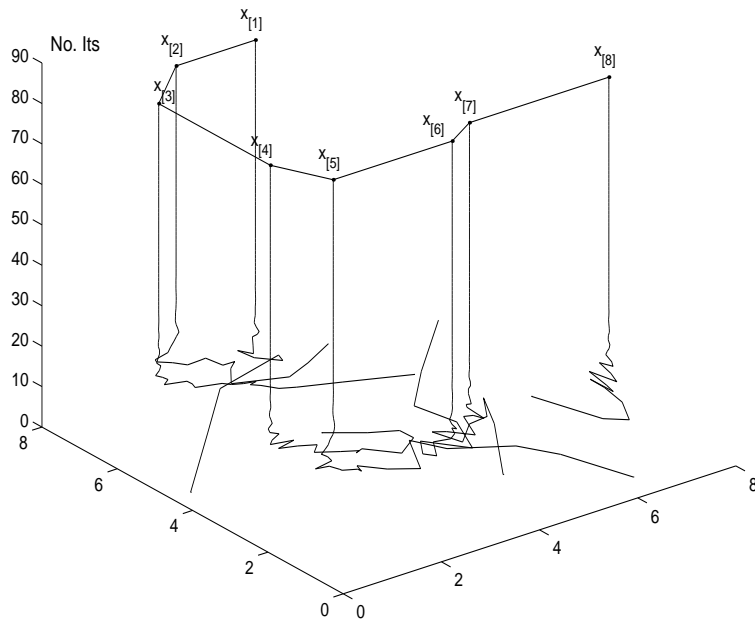


Fig. 7. Convergence tendency, l_1 -norm

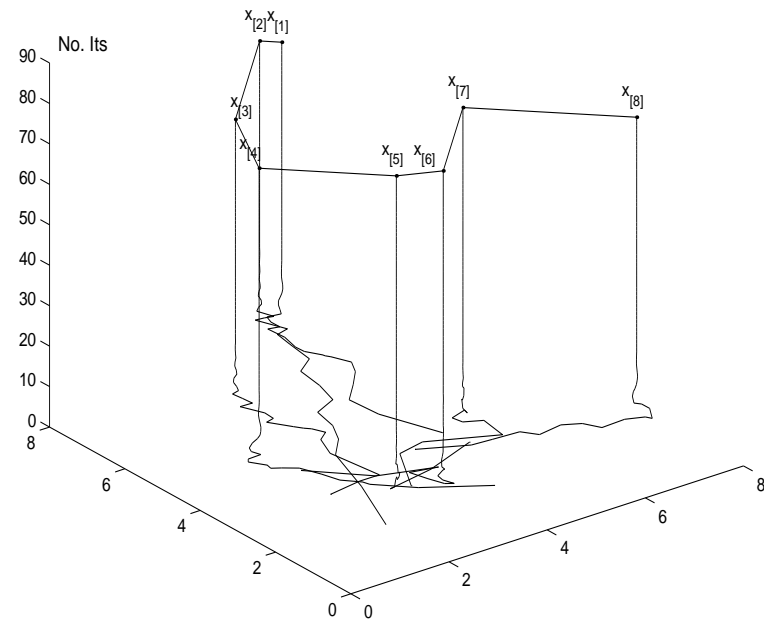


Fig. 8. Convergence tendency, l_∞ -norm

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

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

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

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

从 PC Method-I 到收缩算法 PC Method-II 只是将

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

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'*dx; r2=dx'*dx; 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 $q \in (-500, 500)$

$n =$	PC Method-I			PC Method-II		
	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 $q \in (-500, 0)$

$n =$	PC Method-I			PC Method-II		
	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 $u^k = \tilde{u}^k \Leftrightarrow u^k \in \Omega^*$.

例如, 对给定的 u^k 和 $\beta > 0$, 由投影 $\tilde{u}^k = P_{\Omega}[u^k - \beta F(u^k)]$ 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

$$\tilde{u}^k \in \Omega, \quad (u - \tilde{u}^k)^T d_2(u^k, \tilde{u}^k) \geq (u - \tilde{u}^k)^T d_1(u^k, \tilde{u}^k), \quad \forall u \in \Omega. \quad (5.1a)$$

2. 存在常数 $K > 0$, 使得

$$\|d_1(u^k, \tilde{u}^k)\| \leq K \|u^k - \tilde{u}^k\|. \quad (5.1b)$$

3. 对任意的 $u^* \in \Omega^*$, 有

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

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

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

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

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

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

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

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

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

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

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

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

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 gradient 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*, **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 inequalities in 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] J. Stoer and C. Witzgall, *Convexity and Optimization in Finite Dimension I*. Springer-Verlag, 1970.
- [16] D.F. Sun, A class of iterative methods for solving nonlinear projection equations, *JOTA* **91**, 123-140, 1996.
- [17] 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.
- [18] 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.
- [19] 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.