第 41 卷 第 3 期　　应 用 数 学 学 报　　Vol. 41 No. 3
2018 年 5 月　　ACTA MATHEMATICAE APPLICATAE SINICA　　May, 2018

求解线性互补问题的 Levenberg-Marquardt 型算法*

刘志敏　　杜守强†　　王瑞莹

(青岛大学数学与统计学院，青岛 266071)

(†E-mail: sqdu@qdu.edu.cn)

摘　要　本文考虑了线性互补问题的求解算法，利用一类新的广义互补函数，把线性互补问题转化为非线性方程问题，并且利用 Levenberg-Marquardt 型算法对转化的问题进行了求解．在一般的假设条件下，给出了所给算法的收敛性分析．最后相关的数值结果表明所给的算法十分有效.

关键词　线性互补问题；Levenberg-Marquardt 型算法；全局收敛

MR(2000) 主题分类　90C33

中图分类　O212.2

1　问题介绍

考虑的线性互补问题为：求解 $x \in R^n$, 满足

$$x \geq 0, \qquad Mx+q \geq 0, \qquad x^T(Mx+q)=0, \tag{1}$$

其中 $M \in R^{n\times n}$, $q \in R^n$, 问题 (1) 通常记为 LCP (M,q). 由于线性互补问题在求解工程技术问题、经济和交通均衡问题等许多问题中有着广泛的应用, 线性互补问题解的性质、误差界估计、解集的稳定性及不同类型的互补问题的求解算法等受到了学者的广泛重视, 线性互补问题的理论和算法研究得到了长足发展, 算法设计方面、理论应用与扩展方面的研究成果相当丰硕 [1−14], 其中投影类算法、牛顿法、内点法等算法的研究成果十分丰富. 把线性互补问题转化为非光滑方程, 利用非光滑牛顿法等算法求解线性互补问题也已经得到广泛应用. 转化求解方法的基本思想是通过互补函数把线性互补问题转化为一个与其等价的方程问题, 进而得到互补问题的解. 在转化过程中 Fischer-Burmeister 函数 [1,15]

本文 2017 年 4 月 10 日收到．2018 年 1 月 10 日收到修改稿.

* 国家自然科学基金 (11671220) 和山东省自然科学基金 (ZR2016AM29).

得到广泛的应用, Fischer-Burmeister 函数的定义为 $\phi(a,b)=\sqrt{a^2+b^2}-a-b,\ a,b\in R$. 利用 Fischer-Burmeister 函数, 则问题 (1) 等价于 $\phi_{FB}(x)=0$, 其中 $\phi_{FB}:R^n\to R^n$ 定义为

$$\phi_{FB}(x)=\begin{pmatrix}\phi(x_1,(Mx+q)_1)\\ \vdots\\ \phi(x_n,(Mx+q)_n)\end{pmatrix}.$$

由 [16] 知 $\phi_{FB}(x)$ 在正象限上是相当平坦的, 不能很有效的减少 $x^T(Mx+q)$ 的值, [17,18] 指出利用广义的 Fischer-Burmeister 函数能更好、更灵活的计算互补问题, 并且给出了下降类算法. 本文的动机之一为: 克服 Fischer-Burmeister 函数的上述数值计算缺点, 基于广义 Fischer-Burmeister 函数, 在 [16–18] 的基础上设计更广义的互补转化函数 $\phi_n:R^n\to R^{2n}$,

$$\phi_n(x)=\begin{pmatrix}\lambda\phi_p(x_i,(Mx+q)_i),\quad i=1,\cdots,n\\ (1-\lambda)\phi_+(x_i,(Mx+q)_i),\quad i=1,\cdots,n\end{pmatrix},\tag{2}$$

其中 $\phi_p=\|(a,b)\|_p-(a+b)=\sqrt[p]{|a|^p+|b|^p}-(a+b),\ \ \lambda\in(0,1),\ \ p\in(1,+\infty),\ \ \phi_+(a,b)=a_+b_+,\ \ a_+=\max\{0,a\}$, 则求解问题 (1), 转化为求解 $\phi_n(x)=0$. 另外, 求解形式如 $\phi_n(x)=0$ 的非光滑方程问题是一类十分重要的优化问题 [19−31], 此类问题的研究在算法设计、理论分析等各个方面都得到了长足的发展, 其中 Levenberg-Marquardt 方法由于其适用范围广泛, 计算结构简单得到广泛的应用, 表现出很好的数值稳定性 [25,26]. 同时, 为克服非光滑问题计算的复杂性, 对非光滑问题的光滑近似计算也得到广泛的应用, 利用光滑优化算法对非光滑优化问题设计了很多有效的光滑化梯度类算法等算法, 如 [27–29]. 本文的动机之二为: 对广义 Fischer-Burmeister 函数转化问题 (1), 给出非光滑 Levenberg-Marquardt 型算法, 同时考虑利用光滑化函数近似 (2), 利用光滑化 Levenberg-Marquardt 型算法求解.

具体的函数定义为:

(I) 定义 (2) 的价值函数为

$$\Psi(x)=\frac{1}{2}\|\phi_n(x)\|^2.\tag{3}$$

(II) 定义 (2) 的光滑函数为

$$\phi_n(x,\mu)=\begin{pmatrix}\lambda\phi_p^{\mu}(x_i,(Mx+q)_i),\quad i=1,\cdots,n,\\ (1-\lambda)\phi_+^{\mu}(x_i,(Mx+q)_i),\quad i=1,\cdots,n\end{pmatrix},\tag{4}$$

其中 $\phi_p^{\mu}(a,b)=(|a|^p+|b|^p+\mu)^{\frac{1}{p}}-(a+b),\ \ \phi_+^{\mu}(a,b)=\frac{a+\sqrt{a^2+\mu}}{2}\cdot\frac{b+\sqrt{b^2+\mu}}{2}$, 基于 (4) 的价值函数定义为

$$\tilde{\Psi}(x,\mu)=\frac{1}{2}\|\phi_n(x,\mu)\|^2.\tag{5}$$

本文的结构如下: 第 2 部分给出相关的预备知识, 其中包括半光滑函数, 光滑化函数以及算法中涉及的梯度计算. 第 3 部分结合转化的方程与光滑化的优化问题, 给出求解线性互补问题的两种 Levenberg-Marquardt 型算法并分析了算法的收敛性. 第 4 部分最后

结合算法在具体算例中的应用，表明了算法的有效性．最后结合算法分析，给出了本文的总结．

2　相关预备知识

本部分给出与算法理论分析相关的预备知识，具体可见 [16–18,20,30].

若 $G: R^n \to R^m$ 为局部李普希兹函数， $G'(x)$ 为 G 的雅可比矩阵，则函数 G 在 x 处的 B- 次微分定义为

$$\partial_B G(x) = \big\{V \in R^{m\times n} | \ \exists \{x_k\} \subseteq D_G : \{x_k\} \to x, G'(x_k) \to V\big\},$$

其中 D_G 是可微点集．函数 G 的 Clarke 广义雅可比定义为

$$\partial G(x) = \mathrm{conv}\,\{V \in R^{m\times n} | \ \exists \{x_k\} \subseteq D_G : \{x_k\} \to x, G'(x_k) \to V\}.$$

函数 G 在 x 处的 C- 次微分定义为

$$\partial_C G(x)^T = \partial G_1(x) \times \cdots \times \partial G_m(x).$$

若对 $\forall h \in R^n$, 极限

$$\lim_{V\in\partial G(x+th'),h'\to h,t\to 0^+} Vh'$$

存在，则称 G 在 x 处是半光滑．

定义 2.1　若矩阵 $M \in R^{n\times n}$ 满足

(a)　M 的任意主子式非负，则称 M 是 P_0 矩阵．

(b)　M 的任意主子式为正，则称 M 是 P 矩阵．

定义 2.2　若 $G'(x^*)_{\alpha\alpha}$ 非奇异，并且 $G'(x^*)_{\beta\beta} - G'(x^*)_{\beta\alpha}G'(x^*)^{-1}_{\alpha\alpha}G'(x^*)_{\alpha\beta}$ 是 P 矩阵，则称 x^* 是问题 (1) 的 R- 正则解，其中 $\alpha = \{i|x_i^* > 0,\ G_i(x^*) = 0\}$, $\beta = \{i|x_i^* = 0,\ G_i(x^*) = 0\}$.

定义 2.3　若 $\Psi: R^n \to R^n$ 是局部李普希兹函数， $\tilde{\Psi}: R^n \times R_+ \to R$, 若 $\tilde{\Psi}(\cdot,\mu)$ 在 R^n 上对 $\forall \mu > 0$ 是连续可微的，则对 $\forall x \in R^n$, 有

$$\lim_{\mu\to 0} \tilde{\Psi}(x,\mu) = \Psi(x)$$

成立，则称 $\tilde{\Psi}$ 是 Ψ 的光滑函数．另外，若对 $\forall x_k \in R^n$ 有

$$\lim_{x_k\to x,\mu\to 0} \nabla\tilde{\Psi}(x_k,\mu) \in \partial\Psi(x),$$

则称 Ψ 满足梯度一致性．

命题 2.1　ϕ_P 在点 (a,b) 处的广义梯度定义为

$$\partial\phi(a,b) = (g_a, g_b) = \begin{cases} \Big(\dfrac{\mathrm{sgn}\,(a)|a|^{p-1}}{\|(a,b)\|_p^{p-1}} - 1, \dfrac{\mathrm{sgn}\,(b)|b|^{p-1}}{\|(a,b)\|_p^{p-1}} - 1\Big), & (a,b) \neq (0,0), \\ (\varsigma - 1, \zeta - 1), & (a,b) = (0,0), \end{cases}$$

其中 $\|(\varsigma,\zeta)\|_p \leq 1$. ϕ_+ 在点 (a,b) 处的广义梯度定义为

$$\partial\phi_+(a,b)=\{(b_+\partial a_+, a_+\partial b_+)\},$$

其中

$$\partial z_+=\begin{cases} 1, & z>0\\ [0,1], & z=0\\ 0, & z<0.\end{cases}$$

命题 2.2 ϕ_p^μ 在点 (a,b) 处的广义梯度定义为

$$\begin{aligned}\partial_p^\mu\phi(a,b)=&(g_a^\mu, g_b^\mu)\\ =&\begin{cases}\Big(\dfrac{\mathrm{sgn}\,(a)|a|^{p-1}}{(|a|^p+|b|^p+\mu)^{\frac{p-1}{p}}}-1, \dfrac{\mathrm{sgn}\,(b)|b|^{p-1}}{(|a|^p+|b|^p+\mu)^{\frac{p-1}{p}}}-1\Big), & (a,b)\neq(0,0)\\ (\varsigma-1,\zeta-1), & (a,b)=(0,0),\end{cases}\end{aligned}$$

其中 $\|(\varsigma,\zeta)\|_p \leq 1$. ϕ_+^μ 在点 (a,b) 处的广义梯度定义为

$$\partial\phi_+^\mu(a,b)=\Big\{\Big(\frac{b+\sqrt{b^2+\mu}}{4}\Big(1+\frac{a}{\sqrt{a^2+\mu}}\Big), \frac{a+\sqrt{a^2+\mu}}{4}\Big(1+\frac{a}{\sqrt{b^2+\mu}}\Big)\Big)\Big\}.$$

命题 2.3[30] 设 G 是局部李普希兹函数，且在 x 处是强半光滑的. 若 G 在 x 的某一邻域内方向可导，则

$$\lim_{h\to0,\, H\in\partial G(H+h)}\frac{\|G(x+h)-G(x)-Hh\|}{\|h\|^2}<\infty.$$

命题 2.4 设 x^* 是 (4) 的 R- 正则解, 则任意 $V\in\partial\phi_n(x)$ 非奇异, 且存在 $\alpha>0$, $\beta>0$, 使得对于所有 $x^*\in R^n$, $|x-x^*|\leq\alpha$, $H\in\partial\phi_n(x)$, 有

$$\|(H^TH)^{-1}\|\leq\beta.$$

3 Levenberg-Marquardt 型算法

本部分给出求解问题 (1) 的非光滑 Levenberg-Marquardt 型算法和光滑化 Levenberg-Marquardt 型算法，并且在一般的条件下给出收敛性分析.

首先，给出非光滑 Levenberg-Marquardt 型算法和收敛性分析，其中价值函数定义为

$$\min\Psi(x)=\frac{1}{2}\|\phi_n(x)\|^2.$$

算法 3.1 非光滑 Levenberg-Marquardt 型算法 (NLM).

步 0 令 $\beta\in(0,1)$, $\sigma\in\big(0,\frac{1}{2}\big)$, $\varepsilon\geq0$, $x_0\in R^n$, $k=0$.

步 1 若 $\|\nabla\Psi(x_k)\|\leq\varepsilon$. 则算法终止，否则转步 2.

步 2 选取 $H_k \in \partial_C \phi_n(x_k)$, $\nu_k = \|\nabla\Psi(x_k)\| > 0$, 求解 d_k 满足

$$(H_k^T H_k + \nu_k I)d = -\nabla\Psi(x_k). \tag{6}$$

步 3 计算 $t_k = \max\{\beta^l \mid l = 0, 1, 2, \cdots\}$ 满足

$$\Psi(x_k + t_k d_k) \le \Psi(x_k) + \sigma t_k \nabla\Psi(x_k)^T d_k.$$

令 $x_{k+1} = x_k + t_k d_k$, 转步 1.

引理 3.1 算法 3.1 可行.

证　由 $\nu_k = \|\nabla\Psi(x_k)\| > 0$, 可以得出方程 (6) 有解，即步骤 2 可行．在步骤 3 中，$\Psi(x_k + t_k d_k)$ 在 x_k 处的一阶 Taylor 展式为

$$\Psi(x_k + t_k d_k) = \Psi(x_k) + t_k \nabla\Psi(x_k)^T d_k + o(t_k),$$

将上式代入步骤 3 得

$$\Psi(x_k) + t_k \nabla\Psi(x_k)^T d_k \le \Psi(x_k) + \sigma t_k \nabla\Psi(x_k)^T d_k,$$

由 $\sigma \in \left(0, \frac{1}{2}\right)$, $t_k = \max\left\{\beta^l \mid l = 0, 1, 2, \cdots\right\}$, 得

$$\nabla\Psi(x_k)^T d_k \le 0,$$

即 d_k 是下降方向，所以步骤 3 是可行的，引理成立．证毕.

定理 3.1 由算法 3.1 产生的序列 x_k 的任意聚点都是 Ψ 的稳定点.

证　假设 $\nabla\Psi(x^*) \ne 0$, 令序列 $\{x_k\}$ 收敛到 x^*, 并且子列 $\{x_k\}_K$ 收敛到 x^*, 由 $\{\Psi(x_k)\}$ 的单调下降性和 $\{\Psi(x_k)\}_K$ 收敛到 $\Psi(x^*)$, 得 $\{\Psi(x_k)\}$ 收敛到 $\Psi(x^*)$, 则

$$\lim_{k\to\infty} \Psi(x_{k+1}) - \Psi(x_k) = 0.$$

由算法 3.1 和 d_k 是下降方向，可知

$$\Psi(x_{k+1}) - \Psi(x_k) \le \sigma t_k \nabla\Psi(x_k)^T d_k \le 0.$$

由上述不等式，得

$$t_k \nabla\Psi(x_k)^T d_k \longrightarrow 0. \tag{7}$$

由 (6), 可知

$$t_k \nabla\Psi(x_k)^T d_k = -t_k \nabla\Psi(x_k)^T (H_k^T H_k + \nu_k I)^{-1} \nabla\Psi(x_k). \tag{8}$$

由 C- 次微分的上半连续性知，序列 $\{H_k\}_K$ 有界．我们得到 $\{H_k\}_K \to H^*$, 其中 $H^* \in \partial_C \phi(x^*)$. 由 $\nabla\Psi(x)$ 的连续性可得 $\{\nabla\Psi(x_k)\}_K \to \nabla\Psi(x^*)$. 通过上述讨论，得 $H_k^T H_k + \nu_k I$ 在 K 上收敛到对称正定矩阵 $H^{*T} H^* + \nu^* I$. 由 (7) 和 (8), 可知

$$\{t_k\}_K \to 0.$$

令 l_k 是唯一满足 $t_k=\beta^{l_k}$ 的指数，其中 $k\in N$, 则 β^{l_k-1} 不满足算法 3.1 中的线搜索，因此，对 $\forall k\in K$, 有

$$\frac{\Psi(x_k+\beta^{l_k-1}d_k)-\Psi(x_k)}{\beta^{l_k-1}}>\sigma\nabla\Psi(x_k)^Td_k. \tag{9}$$

由 (6) 得 $\{d_k\}_K\to d^*$, 其中 d^* 满足 $(H^{*T}H^*+\nu^*I)d=-\nabla\Psi(x^*)$. 由 $\{d_k\}_K\to d^*$, $\{x_k\}_K\to x^*$, $\{t_k\}_K\to 0$ 和 (9), 得

$$\nabla\Psi(x^*)^Td^*\geq\sigma\nabla\Psi(x^*)^Td^*.$$

由于 $\sigma\in\left(0,\frac{1}{2}\right)$, 所以 $\nabla\Psi(x^*)^Td^*\geq 0$, 另一方面，由

$$\nabla\Psi(x^*)^Td^*=-\nabla\Psi(x^*)^T(H^{*T}H^*+\nu^*I)^{-1}\nabla\Psi(x^*)<0.$$

矛盾，故假设不成立，所以 x^* 是 Ψ 的稳定点．证毕.

以下给出算法 3.1 收敛速度的分析结果.

定理 3.2 设 $\{x_k\}$ 是由算法 3.1 产生的序列，若 x^* 是 $\{x_k\}$ 的聚点，并且是 R- 正则解，则下面结论成立

(a) 若 $\{\nu_k\}$ 有界，则 $\{x_k\}$ 收敛到 x^*.

(b) 若 $\nu_k\to 0$, 则收敛速度为 Q 超线性.

证 此定理证明与 [30] 中的算法收敛定理证明类似，在此不再给出具体的证明过程．证毕.

注 3.1 R- 正则解与 x^* 有关.

以下给出光滑化 Levenberg-Marquardt 型算法，其中光滑价值函数定义为

$$\min\tilde{\Psi}(x,\mu)=\frac{1}{2}\|\phi_n(x,\mu)\|^2.$$

算法 3.2 光滑化 Levenberg-Marquardt 型算法 (SLM).

步 0 令 $\alpha,\beta\in(0,1),\sigma\in\left(0,\frac{1}{2}\right),\ \mu\in(0,1),\ \gamma>0,\ \varepsilon\geq 0,\ x_0\in R^n,\ k=0$.

步 1 若 $\|\nabla_x\tilde{\Psi}(x_k,\mu_k)\|\leq\varepsilon$, 则算法终止，否则转步 2.

步 2 选取 $H_k\in\partial_x\phi_n(x_k,\mu_k),\ \nu_k=\|\nabla_x\tilde{\Psi}(x_k,\mu_k)\|>0$, 求解 d_k 满足

$$(H_k^TH_k+\nu_kI)d=-\nabla_x\tilde{\Psi}(x_k,\mu_k).$$

步 3 计算 $t_k=\max\{\beta^l\mid l=0,1,2,\cdots\}$ 满足

$$\tilde{\Psi}(x_k+t_kd_k,\mu_k)\leq\tilde{\Psi}(x_k,\mu_k)+\sigma t_k\nabla_x\tilde{\Psi}(x_k,\mu_k)^Td_k.$$

令 $x_{k+1}=x_k+t_kd_k$.

步 4 若 $\|\nabla_x\tilde{\Psi}(x_k,\mu_k)\|\geq\gamma\mu_k$, 则 $\mu_{k+1}=\mu_k$; 否则 $\mu_{k+1}=\alpha\mu_k$, 转步 1.

定理 3.3 若 Ψ 满足梯度一致性，则由算法 3.2 产生序列的任意聚点都是 Clarke 稳定点.

证　定义 $K=\{k \mid \mu_{k+1}=\sigma\mu_k\}$, 若 K 有限，则存在整数 $\bar{k}$, 对 $\forall k>\bar{k}$ 有

$$||\nabla_x \tilde{\Psi}(x_{k+1},\mu_k)|| \geq \gamma\mu_k. \tag{10}$$

记 $\mu_k := \bar{\mu}$, 由于 $\tilde{\Psi}(\cdot,\bar{\mu})$ 为光滑函数，通过算法 3.2 求解 $\min\tilde{\Psi}(x,\bar{\mu})$, 则

$$\liminf_{k\to\infty} ||\nabla_x \tilde{\Psi}(x_{k+1},\bar{\mu})|| = 0.$$

上式与 (10) 矛盾，则 K 无限，并且 $\lim\limits_{k\to\infty}\mu_k=0$. 由 K 中有无穷多个元素，假设 $K=\{k_0,k_1,\cdots\}$, 其中 $k_0<k_1<\cdots$, 得

$$\lim_{i\to\infty} ||\nabla_x \tilde{\Psi}(x_{k_i+1},\mu_{k_i})|| \leq \gamma \lim_{i\to\infty}\mu_{k_i}=0.$$

令 $\bar{x}$ 是 $\{x_{k_i+1}\}$ 的聚点，由梯度一致性可得 $0\in\partial\Psi(\bar{x})$, 因此 $\bar{x}$ 是 Clarke 稳定点．证毕.

4　数值实验

本部分结合文中选取的算例，给出算法 3.1 和算法 3.2 的数值结果，其中包括了 FB 互补函数与 min 函数的情况，具体的数值结果在表 1–12 中给出．同时给出当 $\phi_n(x)$ 用 FB 函数，即 [30] 中算法以及 min 函数定义时所给的 Levenberg-Marquardt 型算法的相关数值比较．ϕ_{FB} 的价值函数定义为

$$\Psi(x)=\frac{1}{2}\|\phi_{FB}(x)\|^2,$$

$\phi_{\min}$ 的价值函数定义为

$$\Psi(x)=\frac{1}{2}\|\phi_{\min}(x)\|^2,$$

其中

$$\phi_{FB}(a,b)=\|(a,b)\|-(a+b), \qquad \phi_{\min}(a,b)=\frac{a+b}{2}-\frac{|a+b|}{2}.$$

所有计算程序代码在 MATLAB R 2013a 环境下运行. 在算法 3.1 中参数 $\beta=0.5,\ \sigma=0.3$, 在算法 3.2 中参数 $\alpha=0.5,\ \beta=0.5,\ \sigma=0.3,\ \gamma=0.5$. 停止规则为 $\|\nabla\Psi(x_k)\|\leq 10^{-6}$ 或者 $k_{\max}=10000$.

在数值结果表格中， DIM 为测试问题的维数， SP 为初始点 x_0, IN 为迭代次数, FV 为 $\Psi(x)=\frac{1}{2}\|\phi(x)\|^2$ 的值， TI 为计算时间，单位为秒．下面给出相关的测试问题，测试问题来源于 [32–34].

例 4.1　(Murty Problem) 矩阵为 P 矩阵，定义为

$$M=\begin{pmatrix} 1 & 2 & 2 & \cdots & 2\\ 0 & 1 & 2 & \cdots & 2\\ 0 & 0 & 1 & \cdots & 2\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}, \qquad q=\begin{pmatrix} -1\\ \vdots\\ -1 \end{pmatrix}.$$

该问题的解为 $x^*=(1,0,1,0)^T$, 该问题在 [32,33] 中也测试过. 在测试中, 选取 $x_0=(1,\cdots,1)^T$ 作为初始点, $\mu_0=0.0001$, $\lambda=0.999$. 算法 3.1 和算法 3.2 测试该问题的数值结果见表 1, 同时算法 3.1 和算法 3.2 测试了用 ϕ_{FB} 和 $\phi_{\min}$ 转化该问题, 分别定义为 $M_1(FB), M_1(\min), M_2(FB), M_2(\min)$, 数值结果见表 2 和表 3.

表 1 算法 3.1 和算法 3.2 求解例 4.1 的数值结果

		算法 3.1			算法 3.2		
DIM	p	FV	IN	TI	FV	IN	TI
200	2	$2.8205e-19$	21	2.2069	$4.8852e-12$	29	1.3712
200	3	$2.8425e-19$	21	1.1164	$9.7702e-12$	28	1.6767
200	4	$2.7589e-19$	21	1.4402	$9.7691e-12$	28	1.9034
200	5	$2.7440e-19$	21	1.2005	$9.7684e-12$	28	1.8926
200	6	$2.7404e-19$	21	1.1992	$1.9538e-11$	27	1.8166
400	2	$2.6984e-15$	27	3.9127	$9.7707e-12$	36	6.7482
400	3	$2.4396e-15$	27	6.2740	$9.7690e-12$	36	9.1647
400	4	$2.3694e-15$	27	6.6079	$1.9539e-11$	35	10.0002
400	5	$2.3520e-15$	27	6.5420	$1.9537e-11$	35	10.0519
400	6	$2.3460e-15$	27	6.6943	$1.9536e-11$	35	9.9637
600	2	$2.5154e-16$	32	11.0390	$7.3269e-12$	42	18.6055
600	3	$2.1754e-16$	32	15.6973	$1.4654e-11$	41	24.0588
600	4	$2.0961e-16$	32	18.6261	$1.4653e-11$	41	26.9884
600	5	$2.0747e-16$	32	17.9435	$2.9307e-11$	40	26.3410
600	6	$2.0666e-16$	32	18.0390	$2.9305e-11$	40	26.4140
800	2	$1.0464e-14$	36	23.5568	$9.7694e-12$	46	38.3408
800	3	$9.0546e-15$	36	32.4932	$1.9539e-11$	45	48.8387
800	4	$8.7107e-15$	36	37.6202	$1.9537e-11$	45	54.1456
800	5	$8.6193e-15$	36	37.1755	$1.9536e-11$	45	53.8584
800	6	$8.5820e-15$	36	37.1800	$3.9075e-11$	44	52.7430
1000	2	$5.0089e-18$	40	42.2822	$1.2212e-11$	49	64.5230
1000	3	$4.2573e-18$	40	61.9488	$1.2210e-11$	50	85.7825
1000	4	$4.0177e-18$	40	66.4500	$2.4422e-11$	49	93.8128
1000	5	$3.9626e-18$	40	66.3485	$2.4420e-11$	49	95.6661
1000	6	$3.9374e-18$	40	65.6294	$2.4419e-11$	49	93.6879
1500	2	$2.2423e-19$	48	156.9537	$1.8319e-11$	58	185.4212
1500	3	$2.1029e-19$	48	168.3035	$1.8316e-11$	58	238.1634
1500	4	$1.8208e-19$	48	191.8120	$3.6635e-11$	57	258.2551
1500	5	$1.8034e-19$	48	198.8395	$3.6632e-11$	57	259.4027
1500	6	$1.7824e-19$	48	189.7050	$3.6630e-11$	57	258.8596

例 4.2[34] 矩阵 M 为三对角矩阵，定义为

$$M=\begin{pmatrix} 4 & -1 & & & & \\ -1 & 4 & -1 & & & \\ & -1 & 4 & -1 & & \\ & & \ddots & \ddots & & \\ & & & & & -1 \\ & & & & -1 & 4 \end{pmatrix}, \qquad q=\begin{pmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{pmatrix}.$$

表 2 当 $\phi_n(x)$ 用 FB 函数定义时算法 3.1 和算法 3.2 求解例 4.1 的数值结果

	$M_1(FB)$			$M_2(FB)$		
DIM	FV	IN	TI	FV	IN	TI
200	$5.2075e-20$	21	0.6607	$2.3215e-27$	29	1.2191
400	$1.8827e-17$	27	3.8511	$4.8977e-27$	36	6.8085
600	$2.8670e-21$	32	11.0143	$4.8939e-28$	41	18.1882
800	$3.2034e-22$	36	23.0111	$6.9767e-28$	45	36.6511
1000	$1.5568e-16$	39	40.4639	$9.5235e-28$	49	64.2524
1500	$1.5600e-13$	46	119.3515	$1.9193e-27$	56	179.9073

表 3 当 $\phi_n(x)$ 用 min 函数定义时算法 3.1 和算法 3.2 求解例 4.1 的数值结果

	$M_1(\min)$			$M_2(\min)$		
DIM	FV	IN	TI	FV	IN	TI
200	$5.4112e-20$	21	0.2613	$1.8920e-30$	28	0.5955
400	$1.8272e-17$	27	1.6844	$1.6301e-29$	35	3.5198
600	$2.7451e-21$	32	4.9580	$4.5914e-30$	40	9.5530
800	$3.0451e-22$	36	11.1501	$1.1531e-29$	44	19.8052
1000	$1.4839e-16$	39	20.3113	$1.6556e-29$	48	35.5948
1500	$1.4939e-13$	46	62.2097	$1.6969e-28$	55	101.3127

在测试中，选取 $x_0=(1,1,\cdots,1)^T$ 作为初始点， $\mu_0=0.001$， $\lambda=0.999$. 算法 3.1 和算法 3.2 测试该问题的数值结果见表 4, 同时算法 3.1 和算法 3.2 测试了用 ϕ_{FB} 和 $\phi_{\min}$ 转化该问题，分别定义为 $M_1(FB), M_1(\min), M_2(FB), M_2(\min)$, 数值结果见表 5 和表 6.

表 4 算法 3.1 和算法 3.2 求解例 4.2 的数值结果

		算法 3.1			算法 3.2		
DIM	p	FV	IN	TI	FV	IN	TI
200	2	$2.0893e-23$	13	0.1787	$3.0331e-13$	23	0.7253
200	3	$5.6207e-19$	13	0.1817	$3.0336e-13$	24	0.5640
200	4	$1.5654e-17$	13	0.2496	$1.5168e-13$	25	0.6341
200	5	$1.6882e-16$	13	0.2388	$7.5849e-14$	26	0.6202
200	6	$1.1323e-15$	13	0.1882	$3.7930e-14$	27	0.4439
400	2	$1.9419e-19$	16	1.4237	$6.0862e-13$	27	2.3794
400	3	$5.5011e-16$	16	1.4013	$3.0430e-13$	28	2.5227
400	4	$7.7188e-15$	16	1.3765	$1.5216e-13$	29	2.6195
400	5	$4.1181e-14$	16	1.3241	$1.5220e-13$	29	2.7242
400	6	$1.4050e-26$	17	1.4429	$7.6108e-14$	30	2.5814
600	2	$2.4122e-14$	18	5.4143	$4.5691e-13$	30	8.6113
600	3	$2.2340e-23$	19	5.6752	$4.5696e-13$	30	8.9487
600	4	$1.1778e-21$	19	5.7393	$2.2848e-13$	31	9.0588
600	5	$1.3441e-20$	19	5.7765	$1.1425e-13$	32	9.1475
600	6	$5.8397e-20$	19	5.7125	$1.1429e-13$	32	9.3419
800	2	$5.4146e-14$	20	13.7725	$6.0954e-13$	32	24.9305
800	3	$6.8958e-23$	21	14.2349	$6.0961e-13$	32	29.8777
800	4	$2.8842e-21$	21	14.6059	$3.0481e-13$	33	26.6657
800	5	$2.7763e-20$	21	14.5268	$1.5241e-13$	34	26.1250
800	6	$1.4098e-19$	21	14.4775	$7.6214e-14$	36	27.2295
1000	2	$3.0474e-16$	22	27.0634	$7.6218e-13$	34	54.1095
1000	3	$1.1855e-26$	23	29.1588	$3.8108e-13$	35	56.0605
1000	4	$8.4941e-25$	23	27.8897	$3.8114e-13$	35	56.7077
1000	5	$8.0953e-24$	23	29.7485	$1.9058e-13$	36	57.4689
1000	6	$3.6017e-23$	23	29.1614	$9.5299e-14$	37	59.8069
2000	2	$1.8144e-17$	29	256.9818	$7.6261e-13$	42	618.8961
2000	3	$1.1703e-14$	29	237.0736	$7.6266e-13$	42	669.1412
2000	4	$1.1099e-13$	29	239.5567	$3.8134e-13$	43	662.5792
2000	5	$7.8650e-26$	30	283.2265	$3.8141e-13$	43	686.4987
2000	6	$4.1695e-25$	30	288.2317	$1.9072e-13$	44	669.4368
4000	2	$7.9970e-23$	39	315.7601	$1.5257e-12$	51	924.9305
4000	3	$3.1971e-19$	39	321.6985	$7.6285e-13$	53	926.6657
4000	4	$5.6959e-18$	39	417.2567	$7.6293e-13$	53	927.2297
4000	5	$2.7199e-17$	39	409.5567	$3.8148e-13$	54	956.9877
4000	6	$7.5917e-17$	39	481.2319	$1.9075e-13$	55	959.8069
6000	2	$1.6713e-18$	46	614.7725	$2.2888e-12$	59	1326.1567
6000	3	$1.2286e-15$	46	627.0975	$1.1444e-12$	60	1457.4689
6000	4	$1.2209e-14$	46	729.1588	$1.1445e-12$	60	1409.3419
6000	5	$4.1705e-14$	46	729.1614	$5.7228e-13$	61	1524.9305
6000	6	$9.1771e-14$	46	783.2265	$2.8616e-13$	62	1557.0605

表 5 当 $\phi_n(x)$ 用 FB 函数定义时算法 3.1 和算法 3.2 求解例 4.2 的数值结果

	$M_1(FB)$			$M_2(FB)$		
DIM	FV	IN	TI	FV	IN	TI
200	$2.0933e-23$	13	0.1766	$1.4902e-27$	26	0.5126
400	$1.9456e-19$	16	1.3400	$1.4616e-28$	30	2.3479
600	$2.4169e-14$	18	5.4152	$2.0501e-28$	33	8.7948
800	$5.4252e-14$	20	13.5833	$5.8967e-28$	35	24.7136
1000	$3.0533e-16$	22	26.4100	$3.2773e-28$	36	56.6716
2000	$1.8179e-17$	29	257.0063	$1.1058e-28$	44	612.9539
4000	$8.0130e-23$	39	405.6675	$2.2146e-28$	54	959.8069
6000	$1.6745e-18$	46	729.1588	$1.0847e-30$	62	1457.4667

表 6 当 $\phi_n(x)$ 用 min 函数定义时算法 3.1 和算法 3.2 求解例 4.2 的数值结果

	$M_1(\min)$			$M_2(\min)$		
DIM	FV	IN	TI	FV	IN	TI
200	$2.1482e-23$	21	0.0961	$1.2202e-27$	25	0.2084
400	$1.2351e-16$	27	0.7527	$1.0833e-28$	29	1.5342
600	$3.5358e-18$	32	2.8388	$1.6022e-28$	32	5.7635
800	$2.0446e-20$	36	8.5427	$2.1145e-28$	34	18.7948
1000	$1.4382e-22$	39	17.6531	$2.6364e-28$	35	41.8222
2000	$2.5586e-29$	46	183.8978	$5.4716e-30$	43	516.9205
4000	$2.2155e-39$	39	315.7656	$8.7826e-30$	53	924.9305
6000	$1.0566e-46$	46	614.7765	$1.0593e-30$	61	1326.1250

例 4.3[34] 矩阵为 M 矩阵，定义为

$$M=\begin{pmatrix} \frac{1}{n} & 0 & 0 & \cdots & 0 \\ 0 & \frac{2}{n} & 0 & \cdots & 0 \\ 0 & 0 & \frac{3}{n} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}, \qquad q=\begin{pmatrix} -1 \\ \vdots \\ -1 \end{pmatrix}.$$

这个问题的解为 $x^*=\left(n,\frac{n}{2},\cdots,1\right)^T$. 在测试中，选取 $x_0=(1,\cdots,1)^T$ 作为初始点，在算法 3.1 中 $\lambda=0.999$, 在算法 3.2 中 $\mu=0.0001$, $\lambda=1$, 算法 3.1 和算法 3.2 测试该问题的数值结果见表 7, 同时算法 3.1 和算法 3.2 测试了用 ϕ_{FB} 和 $\phi_{\min}$ 转化该问题，分别定义为 $M_1(FB), M_1(\min), M_2(FB), M_2(\min)$, 数值结果见表 8 和表 9.

表 7 算法 3.1 和算法 3.2 求解例 4.3 的数值结果

		算法 3.1			算法 3.2		
DIM	p	FV	IN	TI	FV	IN	TI
100	2	$2.7667e-10$	170	0.7479	$4.3115e-26$	175	1.0005
100	4	$9.9947e-10$	176	0.8874	$2.9089e-30$	180	1.1655
100	6	$2.0360e-10$	178	0.0293	$1.2055e-29$	181	1.1656
100	8	$2.7826e-11$	179	0.9715	$1.0287e-28$	181	1.2039
200	2	$4.6321e-10$	346	3.9557	$5.3444e-27$	350	4.3674
200	4	$4.9548e-09$	358	4.0317	$2.8643e-19$	360	4.7731
200	6	$8.9756e-09$	361	4.5572	$4.2552e-28$	364	5.7487
200	8	$9.7249e-11$	363	4.0472	$7.5438e-16$	364	5.1952
400	2	$1.6130e-08$	705	32.3751	$9.4117e-25$	708	35.7996
400	4	$1.8296e-09$	729	34.0647	$3.3431e-19$	731	37.8682
400	6	$3.9876e-09$	734	35.1084	$4.7551e-12$	735	39.2656
400	8	$6.0199e-09$	736	35.8506	$6.0283e-09$	736	38.4199
600	2	$8.8205e-08$	1069	120.2289	$1.5255e-27$	1073	132.6535
600	4	$2.5178e-08$	1103	124.7588	$3.5519e-10$	1104	134.6483
600	6	$1.4365e-08$	1110	124.4162	$1.4383e-08$	1110	136.4583
600	8	$1.0674e-07$	1112	126.9356	$5.0434e-09$	1113	134.6283
800	2	$1.8041e-07$	1436	324.9427	$1.4778e-13$	1439	332.8270
800	4	$1.2769e-07$	1479	331.9708	$1.0606e-08$	1480	341.7942
800	6	$2.1178e-07$	1487	338.1886	$5.7100e-10$	1489	350.3428
800	8	$3.1901e-07$	1490	351.2959	$4.8621e-08$	1491	345.6880
1000	2	$2.6684e-07$	1805	668.4844	$1.2177e-11$	1808	670.0066
1000	3	$1.3020e-07$	1857	663.7329	$3.2264e-10$	1859	700.0671
1000	4	$3.1645e-07$	1866	652.8328	$3.1696e-07$	1866	711.1702
1000	6	$2.2335e-07$	1870	666.1581	$3.5940e-08$	1871	674.3368

表 8 当 $\phi_n(x)$ 用 FB 函数定义时算法 3.1 和算法 3.2 求解例 4.3 的数值结果

	$M_1(FB)$			$M_2(FB)$		
DIM	FV	IN	TI	FV	IN	TI
100	$2.7723e-23$	170	0.8313	$4.3142e-26$	175	1.0004
200	$4.6414e-19$	364	4.1886	$5.4454e-27$	350	4.3674
400	$1.6162e-14$	705	35.3220	$8.6514e-25$	708	35.7995
600	$8.8381e-14$	1069	125.4957	$1.5255e-27$	1073	132.6535
800	$1.8078e-16$	1436	338.9718	$1.4778e-13$	1439	332.8270
1000	$2.6737e-17$	1805	688.7058	$1.2177e-11$	1808	670.0062

表 9 当 $\phi_n(x)$ 用 min 函数定义时算法 3.1 和算法 3.2 求解例 4.3 的数值结果

	M_1(min)			M_2(min)		
DIM	FV	IN	TI	FV	IN	TI
100	$1.3600e-12$	181	0.4769	$3.6232e-29$	184	0.5743
200	$1.3976e-08$	365	2.3473	$1.4942e-28$	369	2.9496
400	$2.4367e-08$	740	21.3814	$7.4674e-15$	742	24.0315
600	$3.6699e-08$	1118	84.7122	$3.1914e-13$	1120	85.2512
800	$1.7758e-07$	1497	223.8941	$3.0517e-10$	1499	233.8619
1000	$4.6822e-07$	1877	483.5934	$1.8882e-10$	1880	485.9840

表 10 算法 3.1 和算法 3.2 求解例 4.4 的数值结果

		算法 3.1			算法 3.2		
DIM	p	FV	IN	TI	FV	IN	TI
200	2	$2.1382e-15$	76	2.8556	$4.7838e-05$	10000	897.5539
200	4	$1.2430e-15$	76	4.9076	$5.1697e-05$	10000	1220.9461
200	6	$1.3659e-13$	76	4.8904	$6.6490e-07$	115	9.4934
200	8	$1.0581e-15$	77	4.9480	$9.0482e-07$	126	10.0207
400	2	$9.9057e-13$	97	15.7483	$3.0056e-04$	173	37.7212
400	4	$8.6044e-14$	98	25.5918	$5.8040e-07$	156	49.5305
400	6	$4.9444e-15$	100	26.0584	$5.1359e-07$	179	56.8116
400	8	$2.8667e-14$	100	26.0811	$1.5798e-06$	163	51.6370
600	2	$3.8241e-15$	113	47.9949	$3.3268e-07$	224	111.2044
600	4	$1.1779e-12$	114	67.8284	$7.1666e-07$	237	168.3005
600	6	$1.7452e-13$	114	67.7657	$3.0980e-06$	247	178.8923
600	8	$1.3703e-12$	114	67.5797	$4.7937e-06$	229	162.2432
800	2	$6.6580e-15$	129	89.9689	$3.3886e-07$	322	292.5678
800	4	$2.0040e-15$	128	141.0305	$8.5292e-07$	357	453.7316
800	6	$1.3517e-14$	127	137.3359	$3.9088e-06$	265	340.1316
800	8	$5.1967e-13$	129	138.0284	$2.9298e-06$	292	372.3228
1000	2	$8.2192e-13$	136	151.5408	$3.4503e-07$	387	554.5820
1000	4	$5.7628e-14$	138	236.1426	$2.1394e-06$	263	530.5368
1000	6	$1.3346e-13$	138	237.9123	$2.2366e-06$	320	642.3367
1000	8	$1.3408e-13$	139	236.9402	$1.6872e-06$	337	683.2777
2000	2	$2.0139e-13$	169	903.0661	$7.1039e-07$	580	3783.1636
2000	4	$2.8693e-13$	171	1305.6566	$3.7597e-06$	899	7951.4485
2000	6	$3.1491e-13$	173	1325.5262	$9.2723e-06$	1101	6791.2894
2000	8	$7.2737e-13$	178	1366.1629	$1.5098e-05$	1011	6071.3644

例 4.4 (Fathi Problem) 矩阵 M 为正定矩阵，定义为

$$
\begin{aligned}
&[M]_{ii} = 4(i-1)+1, \qquad i = 1,\cdots,n;\\
&[M]_{ij} = [M]_{ii}+1, \qquad i = 1,\cdots,n-1, j = i+1,\cdots,n;\\
&[M]_{ij} = [M]_{jj}+1, \qquad j = 1,\cdots,n-1,\ \ i = j+1,\cdots,n;\\
&q = (-1,-1,\cdots,-1)^T.
\end{aligned}
$$

这个问题的解为 $x^* = (1,0,\cdots,0)^T$, 该问题在 [33,34] 中也测试过．在测试中，选取 $x_0 = (1,1,\cdots,1)^T$ 作为初始点， $\mu_0 = 0.00001$, $\lambda = 0.5$. 算法 3.1 和算法 3.2 测试该问题的数值结果见表 10, 同时算法 3.1 和算法 3.2 测试了用 ϕ_{FB} 和 $\phi_{\min}$ 转化该问题，分别定义为 $M_1(FB), M_1(\min), M_2(FB), M_2(\min)$, 数值结果见表 11 和表 12.

表 11 当 $\phi_n(x)$ 用 FB 函数定义时算法 3.1 和算法 3.2 求解例 4.4 的数值结果

	$M_1(FB)$			$M_2(FB)$		
DIM	FV	IN	TI	FV	IN	TI
200	$4.0173e-16$	27	1.3923	$4.7885e-04$	10000	500.1634
400	$4.4164e-15$	38	5.6161	$3.0592e-04$	10000	1889.7653
600	$1.6996e-13$	40	11.8816	$2.0360e-04$	10000	4356.5727
800	$2.8576e-13$	48	30.6562	$1.5447e-04$	10000	7803.0862
1000	$1.0157e-09$	50	42.2213	$1.2400e-04$	10000	11820.3365
2000	$1.9628e-09$	64	275.6994	$1.0407e-04$	10000	10058.1653

表 12 当 $\phi_n(x)$ 用 min 函数定义时算法 3.1 和算法 3.2 求解例 4.4 的数值结果

	$M_1(\min)$			$M_2(\min)$		
DIM	FV	IN	TI	FV	IN	TI
200	$1.1571e-10$	26	1.6738	$5.7669e-04$	10000	321.5739
400	$2.4637e-11$	35	4.0005	$2.9491e-04$	10000	1229.1188
600	$4.7081e-11$	37	5.1185	$1.8073e-04$	10000	2783.8688
800	$4.0444e-11$	45	11.6807	$1.5446e-04$	10000	5324.6480
1000	$2.5117e-11$	48	18.0385	$1.2405e-04$	10000	7881.8423
2000	$1.1954e-11$	62	112.5448	$1.7046e-04$	10000	9821.8673

注 4.1 在 (2) 中， FB 函数是 $\phi_n(x)$ 的一种特殊形式，即 $\lambda = 1$, $p = 2$.

5 总结

Levenberg-Marquardt 算法可以有效的克服牛顿法、拟牛顿法等牛顿类算法中函数雅可比矩阵奇异带来的数值计算不稳定的问题，并且 Levenberg-Marquardt 算法可以有效的求解大规模的数值优化问题，在统计学、经济管理科学等方面有广泛的应用．本文中结合对此类优化问题研究工作的进展，我们基于广义互补函数 ϕ_n, 提出了两种求解

线性互补问题的 Levenberg-Marquardt 型算法，给出了算法的收敛性分析．通过具体的计算可知：ϕ_n 可以灵活的调节 p 的值，提高算法的精度，节省计算时间．最后数值结果表明算法可以有效的解决线性互补问题，并且 ϕ_n 函数比 FB 函数和 min 函数在求解相关问题中更加有效．

致谢． 十分感谢审稿人所提出的宝贵建议．

参 考 文 献

[1] Cottle R W, Pang J S, Stone R E. The Liner Complementarity Problem. New York: Academic, 1992

[2] Zhang L P, Gao Z Y. Global linear and quadratic one-step smoothing Newton method for vertical linear complementarity problems. *Applied Mathematics and Mechanics*, 2003, 24(6): 738–746

[3] Huang Z H, Gu W Z. A smoothing-type algorithm for solving linear complementarity problems with strong convergence properties. *Applied Mathematics and Optimization*, 2008, 57(1): 17–29

[4] Kanzow C. Some noninterior continuation methods for linear complementarity problems. *SIAM Journal on Matrix Analysis and Applications*, 1996, 17(4): 851–868

[5] Burke J V, Xu S. The global linear convergence of a non-interior path-following algorithm for linear complementarity problems. *Mathematics of Operations Research*, 1998, 23(3): 719–734

[6] Fathi Y. Computational complexity of LCPs associated with positive definite matrices. *Mathematical Programming*, 1979, 17(1): 335–344

[7] Wright S. A path-following infeasible-interior-point algorithm for linear complementarity problems. *Mathematical Programming*, 2007, 2(2): 79–106

[8] Balaji R, Palpandi K. On the Lipschitz continuity of the solution map in linear complementarity problems over second-order cone. *Linear Algebra and its Applications*, 2016, 510(2): 146–159

[9] Wu S L, Li C X. Two-sweep modulus-based matrix splitting iteration methods for linear complementarity problems. *Journal of Computational and Applied Mathematics*, 2016, 302: 327–339

[10] Yang W H, Zhang L H, Shen C. Solution analysis for the pseudomonotone second-order cone linear complementarity problem. *Optimization: A Journal of Mathematical Programming and Operations Research*, 2016, 65: 1703–1715

[11] Li C, Li Y. Weakly chained diagonally dominant B-matrices and error bounds for linear complementarity problems. *Numerical Algorithms*, 2016, 73(4): 985–998

[12] Dai P F, Li J C, Li Y T, Bai J C. A general preconditioner for linear complementarity problem with an M-matrix. *Journal of Computational and Applied Mathematics*, 2017, 317: 100–112

[13] Wang S, Zhang K. An interior penalty method for a finite-dimensional linear complementarity problem in financial engineering. Optimization Letters, 2018, in press

[14] Adelgren N, Wiecek M M. A two-phase algorithm for the multiparametric linear complementarity problem. *European Journal of Operational Research*, 2016, 254(3): 715–738

[15] Narushima Y, Sagara N, Ogasawara H. A smoothing Newton method with Fischer-Burmeister function

for second-order cone complementarity problems. *Journal of Optimization Theory and Applications*, 2011, 149(1): 79–101

[16] Kanzow C, Petra S. On a semismooth least squares formulation of complementarity problems with gap reduction. *Optimization Methods and Software*, 2004, 19(5): 507–525

[17] Chen J S. On some NCP-functions based on the generalized Fischer-Burmeister function. *Asia-Pacific Journal of Operational Research*, 2007, 24(3): 401–420

[18] Chen J S, Pan S H. A family of NCP functions and a descent method for the nonlinear complementarity problem. *Computational Optimization and Applications*, 2008, 40(3): 389–404

[19] Fischer A. A special Newton-type optimization method. *Optimization*, 1992, 24: 269–284

[20] Qi L Q, Sun J. A nonsmooth version of Newton method. *Mathematical Programming*, 1993, 58: 353–367

[21] Zhang J L, Wang Y. An new trust region method for nonlinear equations. *Mathematical Methods of Operations Research*, 2003, 58(2): 283–298

[22] Klatte D, Kummer B. Nonsmooth Equations in Optimization. Berlin: Springer, 2002

[23] Qi L Y, Xiao X T, Zhang L W. A parameter-self-adjusting Levenberg-Marquardt method for solving nonsmooth equations. *Computational Mathematics (English Edition)*, 2016(3): 317–338

[24] Huang S, Wan Z. A new nonmonotone spectral residual method for nonsmooth nonlinear equations. *Journal of Computational and Applied Mathematics*, 2016, 313: 82–101

[25] Zhao R X, Fan J Y. Global complexity bound of the Levenberg-Marquardt method. *Optimization Methods and Software*, 2016, 31(4): 805–814

[26] Chen L. A modified Levenberg + Marquardt method with line search for nonlinear equations. *Computational Optimization and Applications*, 2016, 65(3): 1–27

[27] Chen X J. Smoothing methods for nonsmooth, nonconvex minimization. *Mathematical Programming*, 2012, 134(1): 71–99

[28] Zhang L J, Kong L C, Li Y, Zhou S L. A smoothing iterative method for quantile regression with nonconvex l_p penalty. *Journal of Industrial and Management Optimization*, 2017, 13(1): 93–112

[29] Chi X N, Wan Z P, Zhu Z B. The jacobian consistency of a smoothed generalized Fischer-Burmeister function for the second-order cone complementarity problem. *Pacific Journal of Optimization*, 2015, 11(1): 3–27

[30] Facchinei F, Kanzow C. A nonsmooth inexact Newton method for the solution of large-scale nonlinear complementarity problems. *Mathematical Programming*, 1997, 76(3): 493–512

[31] Kanzow C, Petra S. Projected filter trust region method for a semi-smooth least squares formulation of mixed complementarity problems. *Optimization Methods and Software*, 2007, 22: 713–735

[32] Kanzow C. Global convergence properties of some iterative methods for linear complementarity problems. *Optimization*, 2006, 6(2): 326–341

[33] Xu S. The global linear convergence of an infeasible noninterior path-following algorithm for complementarity problems with uniform P-functions. *Mathematical Programming*, 2000, 87(3): 501–517

[34] Geiger C, Kanzow C. On the resolution of monotone complementarity problems. *Computational Optimization and Applications*, 1996, 5(2): 155–173

Levenberg-Marquardt Type Method for Solving Linear Complementarity Problems

LIU Zhimin Du Shouqiang† Wang Ruiying

(*School of Mathematics and Statistics, Qingdao University, Qingdao* 266071)

(† *E-mail: sqdu@qdu.edu.cn*)

Abstract In this paper, we consider the method for solving linear complementarity problems. By a kind of generalized complementarity function, we transform the linear complementarity problems into the nonlinear equations and use the Levenberg-Marquardt type methods to solve it. Under mild conditions, we give the convergence analysis of the given methods. Finally, the numerical results indicate the efficiency of the given methods.

Key words linear complementarity problems; Levenberg-Marquardt type method; global convergence

MR(2000) Subject Classification 90C33

Chinese Library Classification O212.2