

一种求解非线性单调方程组的修正 Liu-Storey 投影算法^①

王松华, 黎 勇, 罗 丹

百色学院 数学与统计学院, 广西 百色 533000

摘要: 针对大规模非线性方程组求解问题, 在 Yuan 研究成果的基础上提出修正的 Liu-Storey 共轭参数公式, 并采用投影技术和一种新型线搜索构建了修正 Liu-Storey 投影共轭梯度算法. 新算法保持了 Yuan 公式不依赖任何线搜索且具有充分下降性的性质, 同时还具有信赖域性质, 在常规条件下新算法具有全局收敛性. 初步的数值试验表明, 新算法总体上比传统的 LS 算法和 3 项 LS 算法更优.

关键词: 非线性方程组; 共轭梯度法; 信赖域性质; 投影技术; 全局收敛性

中图分类号: O224

文献标志码: A

文章编号: 1000-5471(2020)09-0023-08

非线性共轭梯度法具有算法简便, 存储量低的特点, 是解决大规模问题的一种常用而且高效的方法之一, 近来许多研究者将其推广到求解大规模非线性单调方程组问题, 成为优化领域的一个研究热点^[1-7].

本文考虑大规模非线性单调方程组

$$\mathbf{g}(\mathbf{x}) = 0, \mathbf{x} \in \mathbb{R}^n \quad (1)$$

式中 $\mathbf{g}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ 是连续可微的非线性函数; 对 $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, 有如下单调性质:

$$(\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y}))(\mathbf{x} - \mathbf{y}) \geq 0 \quad (2)$$

令范数函数 $\theta(\mathbf{x}) = \frac{1}{2} \|\mathbf{g}(\mathbf{x})\|^2$, 其中 $\|\cdot\|$ 是欧氏范数, 那么非线性单调方程组问题(1)等价于如下全局最优化问题:

$$\min \theta(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n \quad (3)$$

本文基于经典 Liu-Storey 共轭梯度法, 在文献[8]的基础上, 结合文献[1, 7, 9]的方法, 设计一个新的搜索方向公式, 利用文献[10]提出的超平面投影技术和文献[11]一类新型的线搜索, 提出一种求解大规模非线性单调方程组的修正 Liu-Storey 投影算法.

1 修正 Liu-Storey 投影算法

本节扼要介绍共轭梯度法, 然后设计一个新的搜索方向公式, 结合超平面投影技术和某种线搜索方法, 提出一个新 Liu-Storey 投影算法.

共轭梯度法迭代公式为

① 收稿日期: 2019-09-12

基金项目: 国家自然科学基金项目(11661001, 11661009); 广西自然科学基金项目(2018GXNSFAA281259, 2020GXNSFAA159069).

作者简介: 王松华(1970—), 男, 讲师, 主要从事最优化理论与方法的研究.

通信作者: 黎 勇, 教授.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \quad (4)$$

其中: \mathbf{x}_{k+1} 是下一个迭代点, \mathbf{x}_k 是当前迭代点; α_k 是步长, 由某种线搜索确定; \mathbf{d}_k 是搜索方向, 迭代公式定义为

$$\mathbf{d}_k = \begin{cases} -\mathbf{g}_k, & \text{if } k = 1 \\ -\mathbf{g}_k + \beta_k \mathbf{d}_{k-1}, & \text{if } k \geq 2 \end{cases} \quad (5)$$

其中: β_k 是共轭调控参数, \mathbf{g}_k 为 $\mathbf{g}(\mathbf{x}_k)$ 的简写. 经典共轭梯度法有 FR 方法(Fletcher-Reeves)、PRP 方法(Polak-Ribière-Polyak)、CD 方法(Conjugate-Descent) 和 LS 方法(Liu-Storey) 等^[12].

经典的 LS 方法和 PRP 方法在解决大规模无约束优化问题上, 数值结果普遍较好, 然而全局收敛性较差. 而搜索方向的充分下降性和信赖域性质在共轭梯度法全局收敛性中起到关键的作用^[1,2,12-17].

文献[14] 提出一个修正的 PRP 公式, 得到了较好的结果. 其参数公式为

$$\beta_k^{\text{DPRP}} = \beta^{\text{PRP}} - \frac{\mu \|\mathbf{y}_k\|^2}{\|\mathbf{g}_k\|^4} \mathbf{g}_{k+1}^T \mathbf{d}_k$$

其中: 常数 $\mu > \frac{1}{4}$, $\mathbf{y}_{k+1} = \mathbf{g}_{k+1} - \mathbf{g}_k$. 参数公式 β_k^{DPRP} 满足共轭条件和非负性.

文献[8] 在此基础上提出一个新的参数公式为

$$\beta_k^{\text{MPRP}} = \beta^{\text{PRP}} - \min \left\{ \beta^{\text{PRP}}, \frac{\mu \|\mathbf{y}_k\|^2}{\|\mathbf{g}\|^4} \mathbf{g}_{k+1}^T \mathbf{d}_k \right\} \quad (6)$$

β_k^{MPRP} 不仅满足共轭条件和非负性, 而且还具有充分下降性, 在 Wolfe-Powell 线搜索下算法全局收敛, 数值试验结果表明算法有效. 但 β^{MPRP} 公式不具有信赖域性质.

文献[7] 进一步提出修正公式

$$\beta_k^{\text{MPRP}} = \frac{\mathbf{g}_{k+1}^T \mathbf{y}_k - \min \left\{ \mathbf{g}_{k+1}^T \mathbf{y}_k, \frac{\mu \|\mathbf{y}_k\|^2 \mathbf{g}_{k+1}^T \mathbf{d}_k}{\|\mathbf{g}\|^2} \right\}}{\max \left\{ \lambda \|\mathbf{d}_k\| \|\mathbf{y}_k\|, \frac{\gamma \mu \|\mathbf{y}_k\|^2 \|\mathbf{d}_k\|^2}{\|\mathbf{g}_k\|^2} \right\}} \quad (7)$$

其中: 常数 $\mu > \frac{1}{4}$, $\lambda > 0$, $\gamma > 0$. 该公式具有信赖域性质, 克服了文献[8] 的不足. 在适当的条件下证明了算法具有全局收敛性, 数值结果表明方法能有效求解大规模非线性单调方程组.

文献[8] 还给出了具有充分下降性的修正 β^{LS} 公式, 指出在此公式上建立的算法在适当条件下具有全局收敛性的结论, 并给出了该算法的初步数值试验分析, 结果表明该算法比经典 LS 算法更优. 但文献[8] 没有给出算法的搜索方向充分下降性和全局收敛性的证明. 其修正 β^{LS} 公式定义为

$$\beta_k = \beta^{\text{LS}} - \min \left\{ \beta^{\text{LS}}, \frac{\mu \|\mathbf{y}_k\|^2}{(-\mathbf{d}_k^T \mathbf{g}_k)^2} \mathbf{g}_{k+1}^T \mathbf{d}_k \right\}$$

其中: $\beta^{\text{LS}} = \frac{\mathbf{g}_k^T \mathbf{y}_{k-1}}{-\mathbf{g}_{k-1}^T \mathbf{d}_{k-1}}$, $\mathbf{y}_k = \mathbf{g}_k - \mathbf{g}_{k-1}$. 该修正 β 公式也不具有信赖域性质.

本文在文献[8] 的基础上, 结合文献[1,7,9] 关于共轭参数的修正方法, 设计一个新的搜索方向 \mathbf{d}_{k+1} . 此搜索方向不依赖任何线搜索, 不仅具有充分下降性, 还具有信赖域性质. 新的搜索方向 \mathbf{d}_{k+1} 公式定义为

$$\mathbf{d}_{k+1} = \begin{cases} -\mathbf{g}_{k+1} & \text{if } k = 0 \\ -\mathbf{g}_{k+1} + \frac{\mathbf{g}_{k+1}^T \mathbf{y}_k - \min \left\{ \mathbf{g}_{k+1}^T \mathbf{y}_k, \frac{\mu \|\mathbf{y}_k\|^2}{(-\mathbf{d}_k^T \mathbf{g}_k)^2} \mathbf{g}_{k+1}^T \mathbf{d}_k \right\}}{\max \left\{ \lambda \|\mathbf{d}_k\| \|\mathbf{y}_k\|, \frac{\gamma \mu \|\mathbf{y}_k\|^2 \|\mathbf{d}_k\|^2}{(-\mathbf{d}_k^T \mathbf{g}_k)^2} \right\}} \mathbf{d}_k & \text{if } k \geq 1 \end{cases} \quad (8)$$

其中: $\mu > \frac{1}{4}$, $\lambda > 0$, $\gamma > 0$.

为便于表述, 本文将修正 Liu-Storey 投影算法称为 MLS 算法, 其步骤如下:

步骤 1: 给定初始点 $\mathbf{x}_0 \in \mathbb{R}^n$, 常数 $\lambda, \gamma, s > 0$; $\mu > \frac{1}{4}$, $\rho, \varepsilon \in (0, 1)$, 令 $k := 0$;

步骤 2: 若 $\|\mathbf{F}(\mathbf{x}_k)\| \leq \varepsilon$, 则停止, 否则转步骤 3;

步骤 3: 通过式(8) 计算搜索方向 \mathbf{d}_k ;

步骤 4: 利用式(9) 计算步长 α_k

$$-\mathbf{g}(\mathbf{x}_k + \alpha_k \mathbf{d}_k)^\top \mathbf{d}_k \geq \alpha_k \|\mathbf{g}(\mathbf{x}_k + \alpha_k \mathbf{d}_k)\| \|\mathbf{d}_k\|^2 \quad (9)$$

搜索步长 $\alpha_k = \max\{s, \rho s, \rho^2 s, \dots\}$, 常数 $\sigma > 0$, $s > 0$, $\rho \in (0, 1)$.

步骤 5: 利用公式计算下一个迭代点 $\mathbf{z}_k = \mathbf{x}_k + \alpha_k \mathbf{d}_k$;

步骤 6: 若 $\|\mathbf{F}(\mathbf{z}_k)\| \leq \varepsilon$, 则停止, 并令 $\mathbf{x}_{k+1} = \mathbf{z}_k$, 否则, 通过式(10) 计算 \mathbf{x}_{k+1} ,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\mathbf{g}(\mathbf{z}_k)^\top (\mathbf{x}_k - \mathbf{z}_k)}{\|\mathbf{g}(\mathbf{z}_k)\|^2} \mathbf{g}(\mathbf{z}_k) \quad (10)$$

步骤 7: 令 $k := k + 1$, 转步骤 2.

备注: 式(9) 为文献[11] 提出的一种新的单调线搜索技术, 适合应用于非线性单调方程组. 步骤 6 运用了文献[10] 提出的超平面投影技术, 即 \mathbf{x}_{k+1} 是当前迭代点 \mathbf{x}_k 在超平面 $\mathbf{H}_k = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{g}(\mathbf{z}_k)^\top (\mathbf{x} - \mathbf{z}_k) = 0\}$ 上的投影, 其证明如下:

$$\begin{aligned} (\mathbf{x}_{k+1} - \mathbf{x}_k)^\top (\mathbf{x}_{k+1} - \mathbf{z}_k) &= \left(-\frac{\mathbf{g}(\mathbf{z}_k)^\top (\mathbf{x}_k - \mathbf{z}_k)}{\|\mathbf{g}(\mathbf{z}_k)\|^2} \mathbf{g}(\mathbf{z}_k) \right)^\top \left(\mathbf{x}_k - \mathbf{z}_k - \frac{\mathbf{g}(\mathbf{z}_k)^\top (\mathbf{x}_k - \mathbf{z}_k)}{\|\mathbf{g}(\mathbf{z}_k)\|^2} \mathbf{g}(\mathbf{z}_k) \right) = \\ &= \frac{\|\mathbf{g}(\mathbf{z}_k)^\top (\mathbf{x}_k - \mathbf{z}_k) \mathbf{g}(\mathbf{z}_k)\|^2 - \|\mathbf{g}(\mathbf{z}_k)^\top (\mathbf{x}_k - \mathbf{z}_k) \mathbf{g}(\mathbf{z}_k)\|^2}{\|\mathbf{g}(\mathbf{z}_k)\|^2} = 0 \end{aligned}$$

2 充分下降性和信赖域性质

本节讨论 MLS 算法的搜索方向 \mathbf{d}_k 不依赖任何线搜索, 具有充分下降性和信赖域性质.

引理 1 如果搜索方向 \mathbf{d}_k 由式(8) 定义, 则有

$$\mathbf{g}_{k+1}^\top \mathbf{d}_{k+1} \leq -\left(1 - \frac{1}{4\mu}\right) \|\mathbf{g}_{k+1}\|^2 \quad (11)$$

和

$$\left(1 - \frac{1}{4\mu}\right) \|\mathbf{g}_{k+1}\| \leq \|\mathbf{d}_{k+1}\| \leq \left(1 + \frac{1}{\lambda} + \frac{1}{\gamma}\right) \|\mathbf{g}_{k+1}\| \quad (12)$$

引理 1 中式(11)、式(12) 的证明分别类似文献[7,9], 本文不再详述.

3 全局收敛性分析

讨论 MLS 算法的全局收敛性, 需要以下常规性假设 H:

(H1) 非线性方程组(1) 的解集非空, 且水平集 $\Omega = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{g}(\mathbf{x}) \leq \mathbf{g}(\mathbf{x}_1)\}$ 有界.

(H2) 函数 $\mathbf{g}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ 是 Lipschitz 连续的, 即 $\exists L > 0$, 使得

$$\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \quad (13)$$

由假设(H1) 可知, 存在常数 $\omega > 0$, 使得 $\|\mathbf{x}\| \leq \omega$, \mathbf{x}_* 是问题(1) 的解, 即 $\mathbf{g}(\mathbf{x}_*) = 0$, 联合假设(H2) 有

$$\|\mathbf{g}(\mathbf{x})\| = \|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{x}_*)\| \leq L \|\mathbf{x} - \mathbf{x}_*\| \leq L(\|\mathbf{x}\| + \|\mathbf{x}_*\|) \leq 2\omega L$$

这说明存在一个常数 $\vartheta = 2\omega L > 0$, 使得

$$\|\mathbf{g}(\mathbf{x})\| \leq \vartheta \quad (14)$$

首先引入两个重要的引理.

引理 2^[7] 如果假设 H 成立, 序列 $\{\mathbf{x}_k\}$ 由 MLS 算法生成, \mathbf{x}_* 是非线性单调方程组(1) 的解, 那么有

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\|^2 \leq \|\mathbf{x}_k - \mathbf{x}_*\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \quad (15)$$

和

$$\sum_{k=0}^{\infty} \| \mathbf{x}_{k+1} - \mathbf{x}_k \|^2 < \infty \quad (16)$$

成立.

联合超平面投影技术式(10)和线搜索技术式(9), 得到

$$\| \mathbf{x}_{k+1} - \mathbf{x}_k \| = \frac{\alpha_k | \mathbf{g}^T(\mathbf{z}_k) \mathbf{d}_k |}{\| \mathbf{g}(\mathbf{z}_k) \|^2} \| \mathbf{g}(\mathbf{z}_k) \| \geq \alpha_k \| \mathbf{d}_k \|^2$$

从共轭梯度法的迭代公式出发, 通过引理 2 的式(16), 容易得到下面结论:

$$\lim_{k \rightarrow \infty} \alpha_k \| \mathbf{d}_k \| = 0 \quad (17)$$

接下来的引理 3 说明本文所选定的线搜索能够保证得到一个正数步长, 使得 MLS 算法在有限步内停止, 从而说明我们提出的 MLS 算法是有意义的.

引理 3 若假设 H 成立, 则 MLS 算法是适定的.

证 假设 $\| \mathbf{g}_k \| \rightarrow 0$ 不成立, 或者 MLS 算法不停止, 那么 $\exists \eta > 0$, 使得

$$\| \mathbf{g}_k \| \geq \eta, \forall k \in \mathbb{N} \cup \{0\} \quad (18)$$

下面证明满足式(9)的步长 α_k 存在下界.

假设存在 k' 使得式(9)不成立, 那么对 $\forall i \in \mathbb{N} \cup \{0\}$, 令 $\alpha_k^{(i)} = \rho^i s$, 有

$$- \mathbf{g}(\mathbf{x}_{k'} + \alpha_k^{(i)} \mathbf{d}_{k'})^T \mathbf{d}_{k'} < \alpha_k^{(i)} \| \mathbf{g}(\mathbf{x}_{k'} + \alpha_k^{(i)} \mathbf{d}_{k'}) \| \| \mathbf{d}_{k'} \|^2$$

由充分下降性式(11)和假设(H2)有

$$\begin{aligned} \| \mathbf{g}_{k'} \|^2 &\leq - \mathbf{d}_{k'}^T \mathbf{g}_{k'} = \\ &[\mathbf{g}(\mathbf{x}_{k'} + \alpha_k^{(i)} \mathbf{d}_{k'}) - \mathbf{g}(\mathbf{x}_{k'})]^T \mathbf{d}_{k'} - \mathbf{g}(\mathbf{x}_{k'} + \mathbf{d}_{k'})^T \mathbf{d}_{k'} < \\ &[L + \sigma \| \mathbf{g}(\mathbf{x}_{k'} + \alpha_k^{(i)} \mathbf{d}_{k'}) \|] \alpha_k^{(i)} \| \mathbf{d}_{k'} \|^2 \end{aligned} \quad (19)$$

由信赖域性质式(12)和式(14)可得

$$\begin{aligned} \| \mathbf{g}(\mathbf{x}_{k'} + \alpha_k^{(i)} \mathbf{d}_{k'}) \| &\leq \| \mathbf{g}(\mathbf{x}_{k'} + \alpha_k^{(i)} \mathbf{d}_{k'}) - \mathbf{g}(\mathbf{x}_{k'}) \| + \| \mathbf{g}(\mathbf{x}_{k'}) \| \leq \\ &L \alpha_k^{(i)} \| \mathbf{d}_{k'} \| + \vartheta \leq \\ &\left(1 + \frac{1}{\lambda} + \frac{1}{\gamma}\right) L s \vartheta + \vartheta \end{aligned} \quad (20)$$

联合式(12),(19)和(20), 可得

$$\alpha_k^{(i)} > \frac{\| \mathbf{g}(\mathbf{x}_{k'}) \|^2}{[L + \sigma \| \mathbf{g}(\mathbf{x}_{k'} + \alpha_k^{(i)} \mathbf{d}_{k'}) \|] \| \mathbf{d}_{k'} \|^2} > \frac{\eta^2 \lambda^2}{\left[L + \left(1 + \frac{1}{\lambda} + \frac{1}{\gamma}\right) L s \vartheta + \vartheta \right] \left(1 + \frac{1}{\lambda} + \frac{1}{\gamma}\right)^2} > 0$$

这与 $\alpha_k^{(i)}$ 的定义矛盾, 从而说明线搜索(9)能够保证存在一个步长 $\alpha_k > 0$, 使得 MLS 算法在有限步内终止. 因此, 引理 3 成立.

由以上的假设 H、引理 1、引理 2 和引理 3, 可证明下列 MLS 算法的全局收敛性.

定理 1 如果假设 H 成立, 序列 $\{\alpha_k, \mathbf{d}_k, \mathbf{x}_k, \mathbf{g}_k\}$ 由 MLS 算法生成, 则

$$\liminf_{k \rightarrow \infty} \| \mathbf{g}_k \| = 0 \quad (21)$$

证 反证法. 假设式(21)不成立, 则存在一个常数 $\nu > 0$, 使

$$\| \mathbf{g}_k \| \geq \nu, \forall k \in \mathbb{N} \cup \{0\}$$

由信赖域性质式(12), 有

$$\| \mathbf{d}_k \| \geq \| \mathbf{g}_k \| \geq \nu, \forall k \in \mathbb{N} \cup \{0\} \quad (22)$$

由引理 2 序列 $\{\mathbf{x}_k\}$ 有界, 则存在一个聚点 $\tilde{\mathbf{x}}$ 和无限指标集 N_1 , 使得

$$\lim_{k \rightarrow \infty} \mathbf{x}_k = \tilde{\mathbf{x}}, k \in N_1$$

根据式(12)和式(14)得到

$$\| \mathbf{d}_k \| \leq \vartheta \left(1 + \frac{1}{\lambda} + \frac{1}{\gamma} \right), \forall k \in \mathbb{N} \cup \{0\}$$

这说明序列 $\{\mathbf{d}_k\}$ 有界, 所以也存在一个聚点 $\tilde{\mathbf{d}}$ 和无限指标集 N_2 , 使得

$$\lim_{k \rightarrow \infty} \mathbf{d}_k = \tilde{\mathbf{d}}, k \in N_2$$

联合式(17), 可得

$$\lim_{k \rightarrow \infty} \alpha_k = 0$$

由线搜索技术式(9)可得, 存在 $\alpha'_k = \frac{\alpha_k}{\rho}$, 使得

$$-\mathbf{g}(\mathbf{x}_k + \alpha'_k \mathbf{d}_k)^\top \mathbf{d}_k < \alpha \alpha'_k \| \mathbf{g}(\mathbf{x}_k + \alpha'_k \mathbf{d}_k) \| \| \mathbf{d}_k \|^2 \tag{23}$$

当 $k \rightarrow \infty$ 时, 对 $\forall k \in N_2$, (23) 式两边取极限, 即有

$$\mathbf{g}(\tilde{\mathbf{x}})^\top \tilde{\mathbf{d}} > 0$$

而式(11)两边同取极限, 得到

$$\mathbf{g}(\tilde{\mathbf{x}})^\top \tilde{\mathbf{d}} \leq 0$$

显然产生矛盾. 所以假设不成立, 即式(21)成立. 证毕.

4 数值试验

为检验 MLS 算法的有效性, 本节将对 MLS 算法与经典 LS 算法^[12]、3 项 LS 算法^[12] 求解大规模非线性单调方程组问题进行数值试验和比较分析.

选取具有固定初始点的 10 个经典非线性函数^[18-19], 具体的函数名称与初始点见表 1.

表 1 测试问题

序号	问题	\mathbf{x}_0
1	Exponential function1	$\left(\frac{n}{n-1}, \frac{n}{n-1}, \dots, \frac{n}{n-1} \right)$
2	Logarithmic function	$(1, 1, \dots, 1)$
3	Broyden Tridiagonal function	$(-1, -1, \dots, -1)$
4	Trigexp function	$(0, 0, \dots, 0)$
5	Strictly convex function1	$\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n} \right)$
6	Zero Jacobian function	$\left(\frac{(n-1000) * (n-500)}{(60 * n)^2}, \frac{(n-1000) * (n-500)}{(60 * n)^2}, \dots, \frac{(n-1000) * (n-500)}{(60 * n)^2} \right)$
7	Linear function-full rank	$(100, 100, \dots, 100)$
8	Penalty function	$\left(\frac{1}{3}, \frac{1}{3}, \dots, \frac{1}{3} \right)$
9	Extended Powel singular function	$(0.000\ 15, 0.000\ 15, \dots, 0.000\ 15)$
10	Five-diagonal system	$(-2, -2, \dots, -2)$

试验环境: Windows 7 操作系统, Intel Pentium(®) DuaCore CPU E5800 3.2 GHz, 内存 4 GB RAM, MATLA2014a. 终止条件: $\| \mathbf{g}(\mathbf{x}) \| \leq 10^{-5}$ 或 $NI \geq 1\ 000$.

参数设置: $\mu = 0.97$, $\lambda = 0.01$, $\gamma = 0.78$, $\sigma = 0.11$, $\epsilon = 10^{-5}$, 维数为 $[4\ 500, 12\ 000, 24\ 000, 30\ 000, 45\ 000]$. 表 2 给出了 3 种算法的试验结果, 表中 NI 表示迭代次数, NF 表示函数计算的次数, t 表示程序运行所需要的时间.

表 2 数值结果

序号	维数	MLS 算法			LS 算法			3 项 LS 算法		
		NI	NF	t/s	NI	NF	t/s	NI	NF	t/s
1	4 500	719	10 665	4.062 500	530	530	0.828 125	1 000	2 085	2.187 500
	12 000	359	5 272	2.828 125	382	382	1.484 375	1 000	2 114	4.687 500
	24 000	243	3 524	4.156 250	304	304	1.531 250	1 000	2 189	6.671 875
	30 000	218	3 147	4.234 375	282	282	1.625 000	1 000	2 137	7.812 500
	45 000	182	2 605	6.406 250	246	246	1.921 875	1 000	2 180	12.171 875
2	4 500	45	208	0.125 000	63	224	0.171 875	56	270	0.140 625
	12 000	67	347	0.578 125	85	363	0.500 000	78	432	0.515 625
	24 000	100	589	1.062 500	118	602	1.593 750	111	703	1.796 875
	30 000	110	661	1.750 000	128	673	1.890 625	121	784	2.000 000
	45 000	127	782	2.890 625	146	800	3.578 125	138	923	3.671 875
3	4 500	149	1 116	0.312 500	76	288	0.187 500	220	1 632	0.843 750
	12 000	141	948	0.734 375	74	302	0.281 250	242	1 804	1.687 500
	24 000	78	494	0.765 625	93	402	0.609 375	245	1 805	2.812 500
	30 000	115	790	1.468 750	100	449	0.859 375	233	1 707	3.609 375
	45 000	106	794	2.109 375	111	526	1.828 125	269	2 028	5.921 875
4	4 500	65	530	0.328 125	76	501	0.437 500	160	1 364	1.156 250
	12 000	78	687	1.093 750	104	794	1.406 250	196	1 753	3.109 375
	24 000	116	1 061	3.312 500	118	991	2.734 375	235	2 177	6.687 500
	30 000	135	1 269	4.328 125	142	1 214	4.250 000	239	2 250	8.328 125
	45 000	178	1 715	9.531 250	171	1 530	8.781 250	252	2 474	14.562 500
5	4 500	34	158	0.140 625	44	127	0.078 125	63	266	0.078 125
	12 000	53	259	0.468 750	60	226	0.250 000	84	393	0.343 750
	24 000	66	344	0.578 125	75	328	0.703 125	92	489	0.843 750
	30 000	73	392	0.921 875	81	370	0.890 625	97	523	1.125 000
	45 000	88	512	1.593 750	98	498	1.875 000	106	637	2.250 000
6	4 500	209	2 661	0.843 750	270	3 486	0.781 250	221	2 829	0.968 750
	12 000	211	2 709	1.718 750	269	3 485	2.156 250	227	2 904	1.937 500
	24 000	213	2 724	3.171 875	271	3 507	3.953 125	230	2 948	3.546 875
	30 000	214	2 740	3.593 750	273	3 539	4.843 750	226	2 905	4.453 125
	45 000	212	2 722	6.078 125	273	3 533	7.968 750	229	2 926	7.156 250
7	4 500	1 000	14 986	3.906 250	1 000	14 986	4.671 875	1 000	14 986	4.656 250
	12 000	1 000	14 986	10.234 375	1 000	14 986	10.843 750	1 000	14 986	10.312 500
	24 000	1 000	14 986	19.515 625	1 000	14 986	21.171 875	1 000	14 986	17.546 875
	30 000	1 000	14 986	29.562 500	1 000	14 986	25.625 000	1 000	14 986	23.156 250
	45 000	1 000	14 986	39.921 875	1 000	14 986	42.171 875	1 000	14 986	37.609 375
8	4 500	3	17	0.031 250	1 000	14 986	28.390 625	3	17	0.015 625
	12 000	3	17	0.046 875	1 000	14 986	73.125 000	3	17	0.078 125
	24 000	3	17	0.125 000	1 000	14 986	124.812 500	3	17	0.109 375
	30 000	3	17	0.125 000	1 000	14 986	120.234 375	3	17	0.171 875
	45 000	3	17	0.203 125	1 000	14 986	189.859 375	3	17	0.187 500
9	4 500	441	5 948	1.421 875	653	8 776	2.250 000	473	6 288	1.515 625
	12 000	555	7 505	3.890 625	821	11 110	5.093 750	597	7 933	4.171 875
	24 000	642	8 773	8.718 750	968	13 137	12.531 250	699	9 347	8.828 125
	30 000	676	9 239	10.593 750	1 000	13 585	14.750 000	733	9 829	11.062 500
	45 000	746	10 176	19.125 000	1 000	13 471	25.343 750	801	10 765	21.687 500
10	4 500	481	4 751	2.734 375	1 000	6 300	3.781 250	1 000	8 695	4.609 375
	12 000	1 000	10 196	12.093 750	1 000	6 785	9.796 875	1 000	8 250	10.578 125
	24 000	604	6 122	13.843 750	1 000	7 273	17.687 500	1 000	8 711	19.734 375
	30 000	479	4 991	13.703 125	1 000	7 504	21.281 250	1 000	9 086	25.312 500
	45 000	826	8 549	36.187 500	1 000	7 953	34.890 625	365	4 435	18.671 875

从表 2 可以看出, 在相同的条件精度下, MLS 算法在迭代次数 NI 和 CPU 运行时间这两个测试指标上比经典 LS 算法和 3 项 LS 算法好; 在目标函数值计算次数上, MLS 算法与经典 LS 算法有微小差异, 比 3 项 LS 算法更优. 为更直观反映出 MLS 算法、LS 算法和 3 项 LS 算法的性能, 采用文献[20]提出的评价方法, 分别给出这 3 种算法在迭代次数(NI)、目标函数值计算次数(NF)和运行时间等测试结果的性能比较图(图 1—图 3). 这 3 个曲线图表明论文提出的 MLS 算法总体上比经典 LS 算法和 3 项 LS 算法更优, 鲁棒性更好, 适合用来求解大规模非线性单调方程组问题.

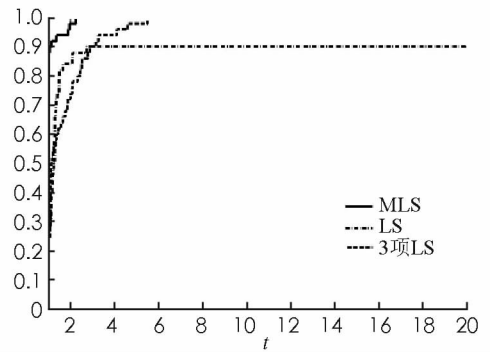


图 1 迭代次数比较图

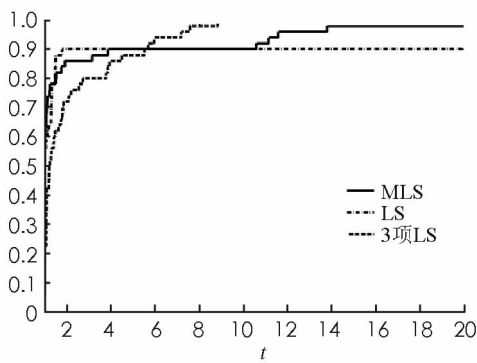


图 2 目标函数计算次数比较图

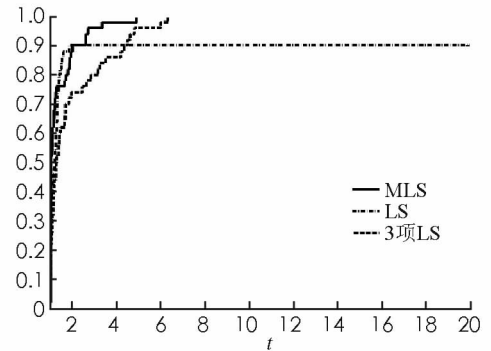


图 3 运行时间比较图

参考文献:

- [1] YUAN G L, WANG B P, SHENG Z. The Hager-Zhang Conjugate Gradient Algorithm for Large-Scale Nonlinear Equations [J]. International Journal of Computer Mathematics, 2019, 96(8): 1533-1547.
- [2] LI X R, WANG X L, SHENG Z, et al. A Modified Conjugate Gradient Algorithm with Backtracking Line Search Technique for Large-Scale Nonlinear Equations [J]. International Journal of Computer Mathematics, 2018, 95(2): 382-395.
- [3] 刘金魁, 杜祥林. 非线性单调方程组的三项无导数投影算法 [J]. 数学进展, 2018, 47(4): 624-634.
- [4] 陈香萍. 谱 HS 投影算法求解非线性单调方程组 [J]. 运筹学学报, 2018, 22(3): 15-27.
- [5] 黎勇. 求解非线性方程组的一种修正 CD 投影算法 [J]. 河南理工大学学报(自然科学版), 2018, 37(6): 155-160.
- [6] 吴晓云, 周学良. 凸约束非线性方程组的一种无导数投影方法 [J]. 数学的实践与认识, 2018, 48(2): 119-126.
- [7] 王博朋, 袁功林, 李春念. 求解非线性方程组的一种修正 WPRP 共轭梯度算法 [J]. 广西大学学报(自然科学版), 2017, 42(5): 1967-1973.
- [8] YUAN G L. Modified Nonlinear Conjugate Gradient Methods with Sufficient Descent Property for Large-Scale Optimization Problems [J]. Optimization Letters, 2009, 3(1): 11-21.
- [9] YUAN G L, WEI Z X, YANG Y N. The Global Convergence of the Polak-Ribière-Polyak Conjugate Gradient Algorithm under Inexact Line Search for Nonconvex Functions [J]. Journal of Computational and Applied Mathematics, 2019, 362: 262-275.
- [10] SOLODOV M V, SVAITER B F. A Globally Convergent Inexact Newton Method for Systems of Monotone Equations [M]// Reformulation; Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods. Boston: Springer US, 1998: 355-369.
- [11] LI Q, LI D H. A Class of Derivative-Free Methods for Large-Scale Nonlinear Monotone Equations [J]. IMA Journal of Numerical Analysis, 2011, 31(4): 1625-1635.

- [12] 戴或虹, 袁亚湘. 非线性共轭梯度法 [M]. 上海: 上海科学技术出版社, 2000.
- [13] 王开荣, 徐晓光. Wolfe 线搜索下充分下降性的 FR 型共轭梯度法 [J]. 西南大学学报(自然科学版), 2017(7): 91-96.
- [14] 喻高航. 大规模优化问题的非线性自调比共轭梯度算法研究 [D]. 广州: 中山大学, 2007.
- [15] 李春念, 袁功林. 求解无约束问题的修正 PRP 共轭梯度算法 [J]. 西南大学学报(自然科学版), 2018, 40(9): 67-75.
- [16] 黎 勇, 韦增欣. 一种自动充分下降的共轭梯度法 [J]. 西南师范大学学报(自然科学版), 2016, 41(5): 36-40.
- [17] 关 哲, 于宪伟. 标准 Wolfe 线搜索下修正的 DY 共轭梯度法 [J]. 西南师范大学学报(自然科学版), 2016, 41(1): 31-34.
- [18] BING Y, LIN G. An Efficient Implementation of Merrill's Method for Sparse or Partially Separable Systems of Nonlinear Equations [J]. SIAM Journal on Optimization, 1991, 1(2): 206-221.
- [19] RAYDAN M. The Barzilai and Borwein Gradient Method for the Large Scale Unconstrained Minimization Problem [J]. SIAM Journal on Optimization, 1997, 7(1): 26-33.
- [20] DOLAN E D, MORE J J. Benchmarking Optimization Software with Performance Profiles [J]. Mathematical Programming, 2002, 91(2): 201-213.

A Modified Liu-Storey Projection Method for Solving Nonlinear Monotone Equations

WANG Song-hua, LI Yong, LUO Dan

School of Mathematics and Statistics, Baise University, Baise Guangxi 533000, China

Abstract: A Liu-Storey conjugate parameter formula proposed by Yuan has been modified on which a modified Liu-Storey projection conjugate gradient method for solving some large-scale nonlinear monotone equations is presented by combining with a new line search technology and projection algorithm. The new algorithm maintains the property that Yuan's formula has a sufficient descent without any line search and has trust region property at the same time, which possesses global convergence in general conditions. Preliminary numerical experiments show that the new algorithm is more efficient than traditional LS algorithm and three-term LS algorithm on the whole. It efficiently solves some large-scale nonlinear monotone equations.

Key words: nonlinear equations; conjugate gradient method; trust region property; projection technique; global convergence

责任编辑 张 桢