

改进的牛顿预测-校正格式

吕 巍^{1†}, 隋瑞瑞¹, 冯恩民²

(1. 上海大学 数学系, 上海 200444; 2. 大连理工大学 数学科学学院, 辽宁 大连 116024)

摘要: 在数值分析领域中, 牛顿算法由于其形式的简单性及快速的收敛性而被广泛地应用于求解非线性方程问题. 受一类求解方程的预测-校正技术的启示, 本文针对求解非线性方程单根的问题提出了一种牛顿预测-校正格式, 并将其推广到多维向量值函数情况. 为此, 首先用图描述了这种新的预测-校正格式并导出了其收敛阶. 这种新格式每步迭代仅需计算一次函数值和一次导函数值. 然后, 经过测试函数的检验, 并与牛顿算法及其他高阶算法(1 + $\sqrt{2}$ 阶、3阶、4阶、5阶、6阶)比较, 表明新算法具有较快的收敛性. 最后, 将这种新格式推广到多维向量值函数, 采用泰勒公式证明了其收敛性, 并给出了一个二维算例来验证其收敛的有效性.

关键词: 牛顿算法; 预测-校正格式; 非线性方程; 迭代方法

中图分类号: TP301.6 **文献标识码:** A

Predictor-corrector improvement of Newton method

LÜ Wei^{1†}, SUI Rui-rui¹, FENG En-min²

(1. Department of Mathematics, Shanghai University, Shanghai 200444, China;

2. School of Mathematical Sciences, Dalian University of Technology, Dalian Liaoning 116024, China)

Abstract: In numerical analysis, Newton method is the most commonly used iterative technique for determining a root of a nonlinear equation for its simplicity and fast rate of convergence. Motivated by a class of predictor-corrector technique for root-finding, we present a predictor-corrector modification for the standard Newton method in approximating the root of a univariate nonlinear function, and extend it to the multi-dimensional vector-valued functions. First, the predictor-corrector rule is described using a figure and its convergence order is analyzed. The modified method only requires evaluating one function and one first derivative in a step. Then, we use numerical examples to demonstrate the faster convergence achieved by this modification than the Newton method and other higher-order (1+ $\sqrt{2}$ order, 3rd order, 4th order, 5th order and 6th order) algorithms. Third, the predictor-corrector improvement is extended to multi-dimensional vector valued functions. The convergence is proved by using Taylor formula, and is illustrated by a two-dimensional example.

Key words: Newton method; predictor-corrector rule; nonlinear equation; iterative methods

1 引言(Introduction)

非线性方程的求解问题是数值分析领域中最为重要的问题之一, 牛顿算法由于其形式的简单性和快速的收敛性而被广泛应用于这类问题中.

设 $f: D \subseteq \mathbb{R}^1 \rightarrow \mathbb{R}^1$ 是有单根 α 的函数, 其中 D 是一个开区间. 那么求解 α 的二阶牛顿算法如下:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

当牛顿算法广泛应用于求解方程的时候, 研究学者提出了许多改进的牛顿算法^[1-9]. 各种改进算法在每步迭代计算函数的数目和收敛阶方面各有优缺点.

比如, 3阶收敛的改进算法^[1-3]和4阶算法^[4]每步迭代需要计算3次函数值. Sharma等^[5]和Dingfang等^[6]的5阶算法每步迭代需要计算4次函数值. Herceg等算法^[7]和House等算法^[8]每步迭代需要计算4次函数值, 均有6阶的收敛速度. 以上这些方法提高了收敛速度, 但每步要计算函数值的次数都多于牛顿法.

2015年, Howk^[9]提出了一类牛顿法的预测-校正格式, 其迭代步骤如下:

$$x_n^* = x_n - \frac{f(x_n)}{\sum_{k=1}^m w_k f'(\lambda_k x_{n-1} + (1 - \lambda_k)x_{n-1}^*)}, \quad (1)$$

收稿日期: 2015-04-03; 录用日期: 2015-06-11.

[†]通信作者. E-mail: lvwei@shu.edu.cn; Tel.: +86 21-66133902.

国家自然科学基金项目(11101262, 11171050), 大连理工大学专项基金(DUTTX2011106), 上海市重点学科资助项目(S30104), 上海高校一流学科(B类)资助.

Supported by National Natural Science Foundation of China(11101262, 11171050), Dalian University of Technology Special Fund(DUTTX2011106), Key Disciplines of Shanghai Municipality(S30104) and a Grant of the First-Class Discipline of Universities in Shanghai.

$$x_{n+1} = x_n - \frac{f(x_n)}{\sum_{k=1}^m w_k f'(\lambda_k x_n + (1 - \lambda_k)x_n^*)}. \quad (2)$$

条件

$$\sum_{k=1}^m w_k = 1 \quad (3)$$

和

$$\sum_{k=1}^m w_k \lambda_k = \frac{1}{2} \quad (4)$$

保证了该算法的收敛阶为 $1 + \sqrt{2}$. 其优点是对于所有的 $\lambda_k \neq 1$, 每步需要计算 $m + 1$ 次函数值.

本文在文献[9]的基础上, 考虑了文献[9]所提出的适用条件(4)以外的情况. 本文取 $m = 1, w_k = 1, \lambda_k = 0$, 此种情况不满足条件(4). 即在文献[9]迭代步骤的预测步(1)中, 取在 x_{n-1}^* 点的导数值; 而在校正步(2)中, 取在 x_n^* 点的导数值. 该算法的优点是每次迭代只需计算 2 个函数值. 通过对一系列数值算例的计算, 以及同牛顿算法和文献[1-9]算法结果的比较, 发现新的预测-校正(predictor-corrector, PC)格式更有效.

2 改进的牛顿算法(The modification method)

本节介绍这种新的预测-校正格式.

在式(1)-(2)中取 $m = 1, w_k = 1, \lambda_k = 0$. 则得到了新的牛顿预测-校正格式:

$$x_n^* = x_n - \frac{f(x_n)}{f'(x_{n-1}^*)}, \quad (5)$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n^*)}. \quad (6)$$

上面的迭代步骤需要两个初值 x_0 和 x_0^* , 在第 1 次迭代搜索 x_1 时取 $x_0^* = x_0$, 从而 $x_1 = x_0 - \frac{f(x_0)}{f'(x_0^*)}$, 即采用牛顿方案. 在搜索 x_2 时, 所取的导数值不是在点 x_1 , 而是点 x_1^* 的值, 而 $x_1^* = x_1 - \frac{f(x_1)}{f'(x_0^*)}$, $f'(x_0^*)$ 在第 1 次迭代时已经得到, 因此在第 2 次迭代时只需计算 $f(x_1)$ 和 $f'(x_1^*)$ 的值. 以上迭代步骤的几何说明如图 1 所示.

因此, 当 $n = 0$ 时, 有

$$\begin{aligned} x_0^* &= x_0, \\ x_1 &= x_0 - \frac{f(x_0)}{f'(x_0^*)}. \end{aligned}$$

当 $n \geq 1$ 时, 有

$$\begin{aligned} x_n^* &= x_n - \frac{f(x_n)}{f'(x_{n-1}^*)}, \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n^*)}. \end{aligned}$$

为了表述方便, 由迭代式(5)和(6)定义的新算法取名为 PC 算法. PC 算法是一种预测-校正格式, 其优点有下面两点:

1) 预测步(5)中 $f'(x_{n-1}^*)$ 的值在上一次迭代中已经得到, 因此 PC(predictor-corrector) 算法每一次迭代仅需要计算两个函数值 $f(x_n)$ 和 $f'(x_n^*)$.

2) 校正步(6)中 x_{n+1} 的值是由函数值 $f(x_n)$ 和 $f'(x_n^*)$ 得到的.

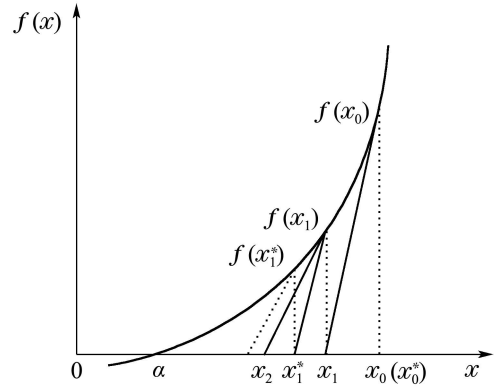


图 1 改进牛顿算法示意图

Fig. 1 Sketch of the modified Newton's method of this paper

3 收敛性分析(Convergence analysis)

本节对 PC 算法的收敛性进行分析, 定理 1 给出了 PC 算法的收敛性并指出新算法的收敛阶是 2.

定理 1 设函数 $f : D \subseteq \mathbb{R}^1 \rightarrow \mathbb{R}^1$ 在开区间 D 上充分光滑, 且有一个单根 $\alpha \in D$.

i) 则由方程(1)和(2)定义的 PC 算法局部收敛到单根 α .

ii) 若 $f'(\alpha) \neq 0, f''(\alpha) \neq 0$, 则 PC 算法的收敛阶为 2.

证 i) PC 算法的局部收敛性证明见文献[9].

ii) 将函数 $f(x)$ 在 α 处 Taylor 展开, 由于 $f(\alpha) = 0$, 可得

$$f(x_n) = f'(\alpha)[e_n + c_2 e_n^2 + O(e_n^3)], \quad (7)$$

其中: $c_k = \frac{1}{k!} f^{(k)}(\alpha)/f'(\alpha), e_n = x_n - \alpha$. 从而

$$f'(x_n^*) = f'(\alpha)[1 + 2c_2 e_n^* + 3c_3 (e_n^*)^2 + O((e_n^*)^3)]. \quad (8)$$

其中 $e_n^* = x_n^* - \alpha$. 用式(7)除以式(8), 可得

$$\frac{f(x_n)}{f'(x_n^*)} = \frac{e_n + c_2 e_n^2 + O(e_n^3)}{1 + 2c_2 e_n^* + 3c_3 (e_n^*)^2 + O((e_n^*)^3)}.$$

根据上述公式和迭代式(5)-(6)可得到如下结果:

$$\begin{aligned} e_0^* &= e_0 \\ e_1 &= c_2 e_0^2 & e_1^* &= 2c_2^2 e_0^3 \\ e_2 &= -c_2^3 e_0^4 & e_2^* &= -2^2 c_2^6 e_0^7 \\ e_3 &= -c_2^7 e_0^8 & e_3^* &= 2^3 c_2^{14} e_0^{15} \\ e_4 &= -c_2^{15} e_0^{16} & e_4^* &= -2^4 c_2^{30} e_0^{31} \\ &\vdots & &\vdots \end{aligned}$$

为了方便分析,对上述无量纲误差化简. 令 $\tilde{e}_i = c_2 e_i$ 和 $\tilde{e}_i^* = c_2 e_i^*$, 则上述结果可化为

$$\begin{aligned} \tilde{e}_0^* &= \tilde{e}_0 \\ \tilde{e}_1 &= \tilde{e}_0^2 & \tilde{e}_1^* &= 2\tilde{e}_0^3 \\ \tilde{e}_2 &= -\tilde{e}_0^4 & \tilde{e}_2^* &= -2^2\tilde{e}_0^7 \\ \tilde{e}_3 &= -\tilde{e}_0^8 & \tilde{e}_3^* &= 2^3\tilde{e}_0^{15} \\ \tilde{e}_4 &= -\tilde{e}_0^{16} & \tilde{e}_4^* &= -2^4\tilde{e}_0^{31} \\ &\vdots & &\vdots \end{aligned}$$

从而可得, 在上述第1列 \tilde{e}_i 的表达式中 \tilde{e}_0 的幂为

$$2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, \dots,$$

记作数列 $\{N_i\} (i = 1, \dots, \infty)$, 根据其数值特点, 可找出其数值间存在的关系式

$$N_i = 2N_{i-1}, \quad i = 2, \dots, \infty. \quad (9)$$

记

$$\lim_{i \rightarrow \infty} \frac{N_i}{N_{i-1}} = \lim_{i \rightarrow \infty} \frac{N_{i-1}}{N_{i-2}} = R.$$

将式(9)两边同时除以 N_{i-1} , 令 $i \rightarrow \infty$ 取极限, 可得 $R = 2$. 综上所述, PC算法的收敛阶为2, 命题得证.

通过以上的分析, PC算法的收敛阶为2. 评价数值算法较好的方法是比较它们的收敛率, 即“效能指数”^[9]. 表1列出了以上所有算法的效能指数. 由于新算法每步迭代需要计算2次函数值, 所以PC算法的效能指数同牛顿算法的效能指数相等, 约为1.4142, 但低于表1中的3阶算法^[1-3]、4阶算法^[4]、5阶算法^[5-6]、6阶算法^[7-8]. 对于 $1 + \sqrt{2}$ 阶算法^[9], 每步迭代计算 $m+1$ 次函数, 下一节的数值实验中分别取 $m = 1$ 和 $m = 2$, 此时算法所对应的能效指标分别为

$$\sqrt[2]{1 + \sqrt{2}} \approx 1.5538 \text{ 和 } \sqrt[3]{1 + \sqrt{2}} \approx 1.3415.$$

下节用数值算例来验证新算法PC的有效性.

表1 各种算法阶和能效指数的比较

Table 1 Comparison of different methods in terms of order and efficiency index

算法	阶	评价指数	效能指数
牛顿算法	2	2	$\sqrt{2} \approx 1.4142$
文献[1-3]算法	3	3	$\sqrt[3]{3} \approx 1.4422$
文献[4]算法	4	3	$\sqrt[3]{4} \approx 1.5874$
文献[5-6]算法	5	4	$\sqrt[4]{5} \approx 1.4953$
文献[7-8]算法	6	4	$\sqrt[4]{6} \approx 1.5651$
文献[9]算法	$1 + \sqrt{2}$	$m + 1$	$\sqrt[m+1]{1 + \sqrt{2}}$
PC算法	2	2	$\sqrt{2} \approx 1.4142$

4 数值算例(Numerical examples)

本节通过数值算例验证新算法PC的有效性. 首先选取14个符合定理1条件的算例, 这些测试函数和18个初值取自于文献[1, 4, 7, 9](见表2); 然后选取方程根

为多重根的算例, 这些算例和初值取自于文献[9](见表3). 这些测试函数涵盖了多项式函数、三角函数、反三角函数、对数函数、指数函数等类型的复合, 具有一定的代表性. PC算法同牛顿算法、 $1 + \sqrt{2}$ 阶算法^[9]、3阶算法^[1-3]、4阶算法^[4]、5阶算法^[5-6]、6阶算法^[7-8]进行了比较. 程序代码运行的平台是Pentium(R)Core (Dural), T4300 CPU, 主频2.10 GHz, 物理内存2.0 GB. 软件环境为MATLAB 7.14 (R2012a), Windows 7旗舰版32位操作系统. 停止准则为 $|x_{n+1} - x_n| < 10^{-27}$ 和 $|f(x_{n+1})| < 10^{-27}$. 针对于每种算法和每个测试函数, 计算了迭代次数, 及每次迭代结束需要计算函数值的数目, 即函数评价指数. 其中: 文献[1]采用的是式(3)(其中 $\alpha = -2$), 记作Z2007; 文献[2]采用的是式(10), 记作BN2013; 文献[3]采用的是式(1), 记作GN2013; 文献[4]采用的是式(17)(19)(24), 分别记作S2012(17), S2012(19), S2012(24); 文献[5]采用的是式(23)–(25), 记作SG2014; 文献[6]采用的是式(12)(其中 $\alpha = 0$), 记作MC2014; 文献[7]采用的是式(7), 记作H2014; 文献[8]采用的是式(14)–(15), 记作2015M2₆; 文献[9]采用的是Trapezoid和Mindpoint方法, 分别记作TR2015和MD2015. 表4和表5分别给出了表2和表3中算例的计算结果, 其中括号外的数字表示迭代次数, 括号里的数字表示函数评价指数, $> 10^3$ 代表迭代次数大于1000, NAN代表算法不收敛.

从表4可以看出, 在252(= 18 × 14)个算例中有246个算例是收敛的. 在全部的收敛算例中, 针对函数评价指数, 算法PC所有数值算例的结果均优于 $1 + \sqrt{2}$ 阶算法TR2015、3阶算法BN2013和6阶算法H2014; PC算法各有17个算例优于3阶算法GN2013、5阶算法SG2014和6阶算法2015M2₆; 15个算例优于5阶算法MC2014; 14个算例优于牛顿算法; 13个算例优于3阶算法Z2007; 12个算例优于4阶算法S2012(19); 11个算例优于4阶算法S2012(24); 9个算例优于4阶算法S2012(17); 6个算例优于 $1 + \sqrt{2}$ 阶算法MD2015. 同时, 在246个收敛的算例中, 针对函数评价指数, PC算法有9个算例与 $1 + \sqrt{2}$ 阶算法MD2015相同; 各有4个算例与牛顿算法, 4阶算法S2012(17), S2012(19)和S2012(24)相同; 有3个算例与3阶算法Z2007相同; 2个算例与5阶算法MC2014相同; 各有1个算例与3阶算法GN2013, 5阶算法SG2014和6阶算法2015M2₆相同. 因此, 同与算法PC同阶的牛顿算法相比, 在18个算例中, 算法PC的函数评价指数有14个优于牛顿算法, 4个和牛顿算法相同; 在比PC算法高阶的12个算法中, 有228个算例收敛, PC算法的函数评价指数有171个算例优于这些算法, 29个算例与这些算法相同. 总之, 在所有收敛的246个算例中, PC算法的函数评价指数有185个算例优于其他算法, 33个算例与其他算法相同.

表 2 测试函数及初值

Table 2 Test functions and initial values

测试函数	初值	根
$f_1(x) = (x - 1)^6 - 1$	$x_0 = 2.5^{[1]}, 3.5^{[1]}$	$\alpha = 2.0$
$f_2(x) = \sin^2 x + x$	$x_0 = 0.2^{[4]}$	$\alpha = 0.0$
$f_3(x) = xe^{-x} - 0.1$	$x_0 = 0^{[4]}$	$\alpha \approx 0.1118325591590$
$f_4(x) = \tan(\ln x) + \sqrt{\frac{1}{2x}} \cos x^3$	$x_0 = 0.4^{[4]}, 0.45^{[4]}$	$\alpha \approx 0.4432607835568$
$f_5(x) = \ln x - x^3 + 2 \sin x$	$x_0 = 2^{[4]}$	$\alpha \approx 1.2979977432804$
$f_6(x) = \cos x + \sqrt{1 - x^2} \sin(2x) + \sin x^2 + x^{14} + x^3 + \frac{1}{2x}$	$x_0 = -0.96^{[4]}$	$\alpha \approx -0.9257722498277$
$f_7(x) = (x - 2)^{23} - 1$	$x_0 = 3.5^{[7]}$	$\alpha = 3.0$
$f_8(x) = e^x - 1.5 + \arctan x$	$x_0 = 1^{[7]}$	$\alpha \approx 0.2369333572389$
$f_9(x) = x^2 - e^x - 3x + 2$	$x_0 = 2^{[9]}, 3^{[9]}$	$\alpha \approx 0.25753028543986$
$f_{10}(x) = x^5 - 10$	$x_0 = 2.5^{[9]}$	$\alpha \approx 1.58489319246111$
$f_{11}(x) = \sin \frac{1}{x} - x$	$x_0 = 3^{[9]}$	$\alpha \approx 0.89753946128049$
$f_{12}(x) = \sin^2 x - x^2 + 1$	$x_0 = 3^{[9]}$	$\alpha \approx 1.4044916482153$
$f_{13}(x) = (x - 1)^3 - 1$	$x_0 = 3.5^{[9]}$	$\alpha = 2.0$
$f_{14}(x) = e^{x^2+7x-30} - 1$	$x_0 = 3.5^{[9]}, 3.25^{[9]}$	$\alpha = 3.0$

表 3 测试函数及初值

Table 3 Test functions and initial values

测试函数	初值	根
$\tilde{f}_1(x) = e^x - x - 1$	$x_0 = 1^{[9]}, 3^{[9]}$	0(二重根)
$\tilde{f}_2(x) = x^2 - 6x + 9$	$x_0 = 0^{[9]}, 4^{[9]}$	3(二重根)
$\tilde{f}_3(x) = \sin x - x$	$x_0 = -1^{[9]}, 3^{[9]}$	0(三重根)

然而, 当方程的根为二重或多重时, 牛顿算法将失去二阶收敛性^[9]. 由文献[9]可知, 当方程的根为二重或多重时, PC算法是收敛的. 表3选取了3个测试函数和6个初值进行验证. 在表3中, $\alpha = 0$ 是 $\tilde{f}_1(x) = 0$ 的二重根, $\tilde{f}'_1(0) = 0, \tilde{f}''_1(0) = 1 > 0$, 因此 $\alpha = 0$ 是函数 $\tilde{f}_1(x)$ 的极小值点; 同理, $\alpha = 3$ 是 $\tilde{f}_2(x) = 0$ 的二重根, $\tilde{f}'_2(0) = 0, \tilde{f}''_2(0) = 2 > 0$, 因此 $\alpha = 3$ 是函数 $\tilde{f}_2(x)$ 的极小值点; 而 $\alpha = 0$ 是 $\tilde{f}_3(x) = 0$ 的三重根, $\tilde{f}'_3(0) = \tilde{f}''_3(0) = 0$, 因此 $\alpha = 0$ 是函数 $\tilde{f}_3(x)$ 的拐点. 从表5可看出, 在84(= 6 × 14)个算例中, 有81个算例是收敛的. 在全部的收敛算例中, 针对函数评价指数, 算法PC有77个算例是最优的, 仅有4个算例差于6阶算法H2014.

综上所述, 从以上的数值结果和分析可以得出, PC算法是有效可行的.

5 进一步讨论(Further discussions)

本节将PC算法推广到多维情况. 设函数 $F(X) : D_l \subseteq \mathbb{R}^l \rightarrow \mathbb{R}^l$ 在凸集 D_l 充分可微, 且只有1个单根 $\tilde{\alpha} = (\alpha_1 \alpha_2 \cdots \alpha_l)^T \in D_l$. 这里:

$$F(X) = (g_1(X), g_2(X), \cdots, g_l(X))^T, \\ X = (X_1, X_2, \cdots, X_l)^T, l \in \mathbb{Z}^+.$$

则PC算法推广的多维迭代格式如下:

$$X_n^* = X_n - [F'(X_{n-1}^*)]^{-1} F(X_n), \quad (10)$$

$$X_{n+1} = X_n - [F'(X_n^*)]^{-1} F(X_n). \quad (11)$$

这里: n 表示迭代次数, $X_n = (X_{n1}, X_{n2}, \cdots, X_{nl})^T$, $F'(X)$ 表示 $F(X)$ 在点 X 的Jacobi矩阵, $[F'(X)]^{-1}$ 表示 $F'(X)$ 的逆矩阵. 其中: 式(10)是预测步, 式(11)是校正步. 上面的迭代步骤需要两个初值 X_0 和 X_0^* , 这里取 $X_0^* = X_0$. 下面的定理2给出了迭代格式(10)和(11)的局部收敛性.

定理 2 设 $F(X) : D_l \subseteq \mathbb{R}^l \rightarrow \mathbb{R}^l$ 在凸集 D_l 充分可微, 且 $F(X) = (g_1(X)g_2(X) \cdots g_l(X))^T, X = (X_1 X_2 \cdots X_l)^T$, 且只有一个单根 $\tilde{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_l)^T \in D_l, l \in \mathbb{Z}^+$. 对于任意小的 $\varepsilon > 0$, 令

$$I_\varepsilon = \{X \in \mathbb{R}^l : \|X - \tilde{\alpha}\|_2 \leq \varepsilon\} \subseteq D_l, \\ F(X) \in C^2(I_\varepsilon).$$

定义

$$\begin{cases} L_{\max} = \max_{X \in I_\varepsilon} \|F'(X)\|_F, \\ L_{\min} = \min_{X \in I_\varepsilon} \|F'(X)\|_F, \\ L'_{\max} = \max_{X \in I_\varepsilon} \|F'(X)^{-1}\|_F, \\ M = \max_{X \in I_\varepsilon} \left\| \frac{1}{2} F''(X) \right\|_F, \\ T = \max\{\|F'(\tilde{\alpha})\|_F - L_{\min}, L_{\max} - \|F'(\tilde{\alpha})\|_F\}, \end{cases} \quad (12)$$

其中: $\|\cdot\|_2$ 表示向量的2-范数, $\|\cdot\|_F$ 表示张量的Frobenius范数. 且当 ε 充分小时, $\tilde{\alpha}$ 是 I_ε 的唯一零点. 若

$$L'_{\max}(T + M\varepsilon) < 1,$$

则对于任意的 $X_0, X_0^*, X_1 \in I_\varepsilon$, 由式(10)和(11)提供的迭代格式收敛到唯一的根 $\tilde{\alpha} \in I_\varepsilon$.

证 令 $X_{n-1}^*, X_{n-1}, X_n \in I_\varepsilon$. 设 $F(X_{n-1})$ 和 $F(X_n)$ 均不为 $\mathbf{0}^T$, 否则迭代格式(10)(11)在有限步内收敛. 下面先证明 $X_n^*, X_{n+1} \in I_\varepsilon$, 再说明 $\lim_{n \rightarrow \infty} X_n = \tilde{\alpha}$. 由式(12)可得

$$\|[F'(X_{n-1}^*)]^{-1}\|_F \leq L'_{\max},$$

记

$$\varepsilon_n^* = X_n^* - \tilde{\alpha} \in \mathbb{R}^l, \varepsilon_n = X_n - \tilde{\alpha} \in \mathbb{R}^l.$$

等式(10)两边同时减去 $\tilde{\alpha}$, 有

$$\varepsilon_n^* = \varepsilon_n - [F'(X_{n-1}^*)]^{-1}F(X_n), \quad (13)$$

将 $F(X_n)$ 在 $X = \tilde{\alpha}$ 处 Taylor 展开, 由 $F(\tilde{\alpha}) = \mathbf{0}^T$, 有

$$\begin{aligned} \|\varepsilon_n^*\|_2 &= \|\varepsilon_n - [F'(X_{n-1}^*)]^{-1}[F'(\tilde{\alpha})\varepsilon_n + \frac{1}{2}F''(\xi)\varepsilon_n^2]\|_2, \end{aligned}$$

其中: ξ 表示点 X_n 与点 $\tilde{\alpha}$ 所连直线内的某一点, $F''(\xi)$ 表示向量值函数 $F''(X)$ 在 ξ 点的张量. 根据 Cauchy 不等式、Minkowski 不等式以及 Frobenius 范数同 2-范数相容的性质, 则式(13)化为

$$\begin{aligned} \|\varepsilon_n^*\|_2 &= \|[F'(X_{n-1}^*)]^{-1}[F'(X_{n-1}^*)\varepsilon_n - F'(\tilde{\alpha})\varepsilon_n - \frac{1}{2}F''(\xi)\varepsilon_n^2]\|_2 \leq \\ &\|[F'(X_{n-1}^*)]^{-1}\|_F \cdot \|[F'(X_{n-1}^*) - F'(\tilde{\alpha})]\|_F \cdot \|\varepsilon_n\|_2 + \\ &\|\frac{1}{2}F''(\xi)\varepsilon_n^2\|_2 \leq \\ &L'_{\max}[\|F'(X_{n-1}^*) - F'(\tilde{\alpha})\|_F + \|\frac{1}{2}F''(\xi)\varepsilon_n\|_F] \cdot \|\varepsilon_n\|_2 \leq \\ &L'_{\max}[T + \|\frac{1}{2}F''(\xi)\|_F \cdot \|\varepsilon_n\|_2] \cdot \|\varepsilon_n\|_2 \leq \\ &L'_{\max}(T + M\varepsilon)\|\varepsilon_n\|_2 \leq \\ &L'_{\max}(T + M\varepsilon)\varepsilon < \varepsilon. \end{aligned}$$

因此 $X_n^* \in I_\varepsilon$. 同理可得

$$\|\varepsilon_{n+1}\|_2 \leq L'_{\max}(T + M\varepsilon)\|\varepsilon_n\|_2 < \varepsilon,$$

从而 $X_{n+1} \in I_\varepsilon$, 且有

$$\|\varepsilon_{n+1}\|_2 \leq (L'_{\max}(T + M\varepsilon))^{n+1}\|\varepsilon_0\|_2,$$

当 $n \rightarrow \infty$ 时, $\|\varepsilon_{n+1}\|_2 \rightarrow 0$, 因此迭代格式(10)-(11)是局部收敛的, 命题得证.

在定理2中, 当 l 取 1 时, 式(10)和(11)即为式(5)和(6), 因此迭代格式(10)和(11)是式(5)和(6)的推广. 总之, PC 算法的使用范围由于新格式的推广而被扩大, 对于一维收敛 PC 格式(5)和(6)的使用范围是要求函数在某开区间上足够光滑即可; 而对于多维收敛格式(10)和(11), 其使用范围即为定理2的条件.

下面举一多维实例来验证迭代式(10)和(11)的收敛性.

例 取 $l = 2, D_2 = (-2, 2) \times (-2, 2), X = (X_1, X_2)^T, F(X) = (g_1(X_1, X_2), g_2(X_1, X_2))^T = (X_1, X_2)^T$, 有单根 $\tilde{\alpha} = (0, 0)^T$.

i) 首先来验证向量值函数 $F(X)$ 满足定理2的条件. 显然 $F(X)$ 在凸集 D_2 上任意阶可微. 并且对任意小的 $\varepsilon > 0, I_\varepsilon = \{X \in \mathbb{R}^2 : \sqrt{X_1^2 + X_2^2} \leq \varepsilon\} \subseteq D_2$.

$$F'(X) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, F'(X)^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$F''(X) = \mathbf{0}_{2 \times 2 \times 2},$$

从而

$$L_{\max} = \max_{X \in I_\varepsilon} \|F'(X)\|_F = \sqrt{2},$$

$$L_{\min} = \min_{X \in I_\varepsilon} \|F'(X)\|_F = \sqrt{2},$$

$$L'_{\max} = \max_{X \in I_\varepsilon} \|F'(X)^{-1}\|_F = \sqrt{2},$$

$$M = \max_{X \in I_\varepsilon} \|\frac{1}{2}F''(X)\|_F = 0,$$

$$T = \max\{\|F'(\tilde{\alpha})\|_F - L_{\min}, L_{\max} - \|F'(\tilde{\alpha})\|_F\} = 0,$$

从而

$$L'_{\max}(T + M\varepsilon) = \sqrt{2} \cdot (0 + 0 \cdot \varepsilon) = 0 < 1$$

满足定理2条件.

ii) 下面验证式(10)和(11)的收敛性. 对任意小的 $\varepsilon > 0$, 取 $X_0^* = X_0 = (\varepsilon, 0)^T \in I_\varepsilon$, 则

$$\begin{aligned} X_1 &= X_0 - [F'(X_0^*)]^{-1}F(X_0) = \\ &(\varepsilon, 0)^T - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} (\varepsilon, 0)^T = (0, 0)^T = \tilde{\alpha}, \end{aligned}$$

所以向量值函数 $F(X)$ 经过式(10)和(11)迭代收敛到单根 $\tilde{\alpha}$.

6 结论(Conclusions)

本文针对具有单根的非线性方程求解问题, 提出了一种阶为2的牛顿预测-校正格式, 并将其推广到了多维格式. 针对函数根为单根和重根的情况, 分别给出测试函数验证算法的有效性, 并将 PC 算法的函数评价指数同 13 种算法的结果做了比较. 在全部 336 (= 24 × 14) 个算例中, 关于函数评价指数, PC 算法有 266 个算是最优值, 有 33 个算例与其他算法相等. 结果表明: 改进的 PC 算法的函数评价指数均优于其余 13 种算法. 因此算法 PC 可以作为文献 [9] 的一种补充.

此外, 多维格式的 PC 算法只考虑了向量值函数 $F(X) : D_l \subseteq \mathbb{R}^l \rightarrow \mathbb{R}^l$, 对于更一般的情况以及针对于方程有多重根时一元格式 PC 算法的阶, 将在以后的工作中做进一步的探讨.

表 4 方程有单根时, PC 算法同其他算法^[1-9] 的数值结果比较
Table 4 Numerical comparisons of algorithms^[1-9] and PC with functions having a simple root

测试函数	x_0	牛顿算法	MD2015 ^[9]	TR2015 ^[9]	Z2007 ^[1]	BN2013 ^[2]	GN2013 ^[3]	S2012(17) ^[4]	S2012(19) ^[4]	S2012(24) ^[4]	SG2014 ^[5]	MC2014 ^[6]	H2014 ^[7]	2015M ₂₆ ^[8]	PC 算法
$f_1(x)$	2.5	9(18)	8(16)	8(24)	5(15)	6(18)	6(18)	6(18)	6(18)	5(15)	5(20)	5(20)	6(24)	5(20)	8(16)
	3.5	12(24)	10(20)	10(30)	7(21)	7(21)	8(24)	5(15)	7(21)	7(21)	6(24)	5(20)	7(28)	6(24)	10(20)
$f_2(x)$	0.2	7(14)	6(12)	6(18)	5(15)	5(15)	5(15)	4(12)	4(12)	4(12)	4(16)	4(16)	6(24)	4(16)	6(12)
$f_3(x)$	0	6(12)	6(12)	6(18)	4(12)	5(15)	4(12)	4(12)	4(12)	4(12)	4(16)	4(16)	6(24)	4(16)	6(12)
$f_4(x)$	0.4	7(14)	6(12)	6(18)	4(12)	6(18)	6(18)	4(12)	4(12)	4(12)	4(16)	4(16)	6(24)	4(16)	6(12)
	0.45	6(12)	5(10)	5(15)	4(12)	5(15)	5(15)	4(12)	4(12)	4(12)	3(12)	3(12)	5(20)	3(12)	6(12)
$f_5(x)$	2	8(16)	7(14)	7(21)	5(15)	5(15)	5(15)	5(15)	5(15)	5(15)	4(16)	4(16)	6(24)	4(16)	7(14)
$f_6(x)$	-0.96	7(14)	6(12)	6(18)	4(12)	5(15)	5(15)	4(12)	4(12)	4(12)	4(16)	4(16)	6(24)	4(16)	7(14)
$f_7(x)$	3.5	16(32)	14(28)	14(42)	9(27)	10(30)	10(30)	10(30)	10(30)	9(27)	8(32)	> 10 ³	8(32)	8(32)	13(26)
$f_8(x)$	1	7(14)	6(12)	6(18)	5(15)	5(15)	5(15)	5(15)	5(15)	5(15)	6(24)	4(16)	4(16)	4(16)	6(12)
$f_9(x)$	2	6(12)	6(12)	6(18)	5(15)	5(15)	5(15)	24(72)	24(72)	27(81)	4(16)	4(16)	6(24)	4(16)	6(12)
	3	8(16)	7(14)	7(21)	5(15)	6(18)	6(18)	NAN	7(21)	8(24)	5(20)	> 10 ³	7(28)	4(16)	7(14)
$f_{10}(x)$	2.5	9(18)	8(16)	8(24)	5(15)	6(18)	6(18)	NAN	5(15)	5(15)	5(20)	4(16)	6(24)	5(20)	7(14)
$f_{11}(x)$	3	8(16)	6(12)	6(18)	5(15)	5(15)	6(18)	5(15)	NAN	NAN	4(16)	4(16)	7(28)	4(16)	7(14)
$f_{12}(x)$	3	8(16)	7(14)	7(21)	5(15)	5(15)	5(15)	5(15)	6(18)	6(18)	5(20)	4(16)	7(28)	11(44)	6(12)
$f_{13}(x)$	3.5	9(18)	8(16)	8(24)	5(15)	6(18)	6(18)	5(15)	5(15)	5(15)	5(20)	4(16)	6(24)	5(20)	7(14)
$f_{14}(x)$	3.5	14(28)	12(24)	12(36)	8(24)	8(24)	9(27)	8(34)	8(34)	8(34)	7(28)	> 10 ³	7(28)	7(28)	11(22)
	3.25	10(20)	9(18)	9(27)	6(18)	6(18)	7(21)	6(18)	6(18)	6(18)	5(20)	NAN	7(28)	5(20)	8(16)

表 5 方程有多重根时, PC 算法同其他算法^[1-9] 的数值结果比较
Table 5 Numerical comparisons of algorithms^[1-9] and PC with the roots of functions having multiplicity

测试函数	x_0	牛顿算法	MD2015 ^[9]	TR2015 ^[9]	Z2007 ^[1]	BN2013 ^[2]	GN2013 ^[3]	S2012(17) ^[4]	S2012(19) ^[4]	S2012(24) ^[4]	SG2014 ^[5]	MC2014 ^[6]	H2014 ^[7]	2015M ₂₆ ^[8]	PC 算法
$\tilde{f}_1(x)$	1	91(182)	75(150)	75(225)	41(123)	58(174)	52(156)	53(159)	53(159)	51(153)	45(180)	38(152)	28(112)	41(164)	61(122)
	1	93(186)	77(154)	77(231)	43(129)	59(177)	54(162)	54(162)	54(162)	52(156)	47(188)	39(156)	29(116)	42(168)	63(126)
$\tilde{f}_2(x)$	0	92(184)	76(152)	76(228)	42(136)	58(174)	53(159)	NAN	NAN	NAN	46(184)	39(156)	28(112)	42(168)	62(124)
	4	90(180)	75(150)	75(225)	41(123)	58(174)	52(156)	53(159)	53(159)	51(153)	45(180)	39(156)	27(108)	41(164)	61(122)
$\tilde{f}_3(x)$	-1	152(304)	126(252)	127(381)	72(216)	101(301)	90(270)	89(267)	89(267)	85(255)	75(300)	54(216)	54(216)	67(268)	105(210)
	3	154(308)	128(256)	128(384)	73(219)	101(301)	91(273)	> 10 ³	> 10 ³	> 10 ³	76(304)	54(216)	54(216)	68(272)	107(214)

参考文献(References):

- [1] ZHOU X. A class of Newton's methods with third-order convergence [J]. *Applied Mathematics Letters*, 2007, 20(9): 1026 – 1030.
- [2] ARDELEAN G. A new third-order Newton-type iterative method for solving nonlinear equations [J]. *Applied Mathematics and Computation*, 2013, 219(18): 9856 – 9864.
- [3] HERCEG DJ, HERCEG DR. Third-order modifications of Newton's method based on Stolarsky and Gini means [J]. *Journal of Computational and Applied Mathematics*, 2013, 245: 53 – 61.
- [4] SOLEYMANI F, KHATTRI S K, VANANI S K. Two new classes of optimal Jarratt-type fourth-order methods [J]. *Applied Mathematics Letters*, 2012, 25(5): 847 – 853.
- [5] SHARMA J R, GUPTA P. An efficient fifth order method for solving systems of nonlinear equations [J]. *Applied Mathematics and Computation*, 2014, 67(3): 591 – 601.
- [6] DINGFANG L, PING L, JISHENG K. An improvement of Chebyshev – Halley methods free from second derivative [J]. *Computers and Mathematics with Applications*, 2014, 235: 221 – 225.
- [7] HERCEG DJ, HERCEG DR. Sixth-order modifications of Newton's method based on Stolarsky and Gini means [J]. *Journal of Computational and Applied Mathematics*, 2014, 267: 244 – 253.
- [8] HUESO J L, MARTÍNEZ E, TERUEL C. Convergence, efficiency and dynamics of new fourth and sixth order families of iterative methods for nonlinear systems [J]. *Journal of Computational and Applied Mathematics*, 2015, 275: 412 – 420.
- [9] HOWK C L. Convergence of a class of efficient quadrature-based predictor-corrector methods for root-finding [J]. *Applied Mathematics and Computation*, 2015, 252: 189 – 200.

作者简介:

吕巍 (1978–), 女, 博士, 讲师, 目前研究方向为分布参数系统的参数辨识及其应用等研究, E-mail: lvwei@shu.edu.cn;

隋瑞瑞 (1992–), 女, 硕士研究生, 目前研究方向为分布参数系统参数辨识的算法研究, E-mail: suiruirui1992@163.com;

冯恩民 (1939–), 男, 教授, 博士生导师, 目前研究方向为集中与分布参数系统的参数辨识与最优控制及其应用等研究, E-mail: emfeng@dlut.edu.cn.

更正

2014年第31卷第1期第39页的图7有误, 更正如下:

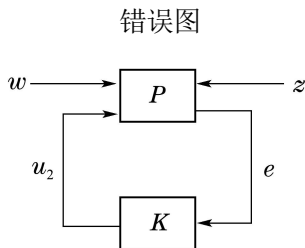


图7 标准 H_∞ 控制问题框图

Fig. 7 The block diagram of the standard H_∞ control

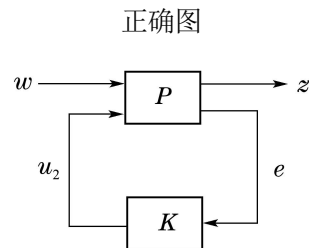


图7 标准 H_∞ 控制问题框图

Fig. 7 The block diagram of the standard H_∞ control

特此更正. 并向作者和读者表示诚挚的歉意!