

A NOVEL AND PRECISE SEVENTH-ORDER NEWTON'S ITERATIVE METHOD FOR SOLVING NONLINEAR EQUATIONS

WANG Xiao-feng^{1,2}, SHI Dong-yang¹

(*1.School of Math. and Statistics, Zhengzhou University, Zhengzhou 450001, China*)

(*2.Department of Math., Henan Institute of Science and Technology, Xinxiang 453003, China*)

Abstract: In this paper, we study the problem of solving nonlinear equations. By using Taylor formulas and coupling method, we get a novel and robust three-step seventh-order iterative scheme. The contributed without memory method does not need to calculate higher order derivatives and has a large radius of convergence and higher efficiency of calculation.

Keywords: iterative method; nonlinear equations; extended computational index; efficiency index; convergence radius

2010 MR Subject Classification: 65H04; 65H05

Document code: A

Article ID: 0255-7797(2015)05-1017-09

1 Introduction

In numerical analysis and scientific work, one of the most important occurring problems is solving nonlinear equations. To solve these equations, we can use iterative methods such as Newton's method and its variants. In this paper, we consider iterative methods to find a real simple root x^* of a nonlinear equation $f(x)=0$, where $f : D \rightarrow R$ is a scalar function and it is sufficiently smooth in a neighborhood of x^* , and D is an open interval.

Being quadratically convergent, Newton's method is probably the best known and most used algorithm. To improve the local order of convergence of Newton's method, many modified methods were proposed. One of the third-order modifications of Newton's method [1] was given by

$$x_{n+1} = x_n - \frac{f(x_n) + f(y_n)}{f'(x_n)}, \quad (1.1)$$

where $y_n = x_n - f(x_n)/f'(x_n)$, $n \geq 0$. Also, another third-order method [2], that is defined by

$$x_{n+1} = x_n - \frac{f^2(x_n)}{f(x_n) - f(y_n)} \cdot \frac{1}{f'(x_n)}. \quad (1.2)$$

* **Received date:** 2013-04-17

Accepted date: 2014-07-31

Foundation item: Supported by National Natural Science Foundation of China (U1304106).

Biography: Wang Xiaofeng (1977–), male, Mongolian, born at Neixiang, Henan, associated professor, major in numerical solution of partial differential equations and numerical algebra.

Combining eqs. (1.1) and (1.2), we have

$$x_{n+1} = \alpha \left[x_n - \frac{f(x_n) + f(y_n)}{f'(x_n)} \right] + \beta \left[x_n - \frac{f^2(x_n)}{f(x_n) - f(y_n)} \cdot \frac{1}{f'(x_n)} \right].$$

Taking $\alpha = -1, \beta = 2$, we can obtain the following fourth-order iterative method [3]

$$x_{n+1} = x_n - \frac{f^2(x_n) + f^2(y_n)}{f'(x_n)[f(x_n) - f(y_n)]}. \quad (1.3)$$

The present method requires two evaluations of the function and one evaluation of its first derivative. We consider the definition of efficiency index [3] as $p^{1/m}$, where p is the order of the method and m is the number of function evaluations in per iteration required by the method. The present methods defined by (1.3) has the efficiency index equals to $\sqrt[3]{4} \approx 1.587$, which is better than the ones of the mentioned third-order methods $\sqrt[3]{3} \approx 1.442$ and Newton's method $\sqrt{2} \approx 1.414$.

In order to derive more new high-order iterative methods, we present and analyze a new three-step iterative method by approximating the first derivative of the function in the third step for solving nonlinear equations. According to the next theorem, the local order of convergence of the composed method will be at least seven and one iteration this method requires three evaluations of the function and one evaluation of its first derivative with its efficiency index $\sqrt[4]{7} \approx 1.627$. Several numerical examples are given to show the performance of the new method.

2 Description of the Method

In this section, for construction of the new high-order iterative method, we use the iterative method given by eq. (1.3).

Theorem 2.1 Let $\psi_1(x)$ and $\psi_2(x)$ be two iterative methods with order of convergence P and Q , respectively, then the order of convergence of the iterative method $\psi(x) = \psi_2(\psi_1(x))$ is PQ [4].

We consider the iteration scheme of the form

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \frac{f^2(x_n) + f^2(y_n)}{f'(x_n)[f(x_n) - f(y_n)]}, \\ x_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}. \end{cases} \quad (2.1)$$

From Theorem 2.1, we can see that the above formulas have an efficiency index of $\sqrt[5]{8} \approx 1.516$. To improve the efficiency index, we approximate the first-appeared derivative in the last step $f'(z_n)$ by a combination of already evaluated function values using divided differences.

We consider the Taylor polynomial of degree 2 for the function $f(z_n)$:

$$f(z_n) \approx f(y_n) + f'(y_n)(z_n - y_n) + \frac{1}{2}f''(y_n)(z_n - y_n)^2, \quad (2.2)$$

which implies

$$f'(y_n) \approx f[z_n, y_n] - \frac{f''(y_n)}{2}(z_n - y_n). \quad (2.3)$$

But $f''(y_n)$ can be expressed as [4]

$$f''(y_n) \approx \frac{2f[z_n, x_n] - 2f'(x_n)}{z_n - x_n} = 2f[z_n, x_n, x_n]. \quad (2.4)$$

It is easy to know that

$$f'(z_n) \approx f[z_n, y_n] + f[z_n, x_n, x_n](z_n - y_n). \quad (2.5)$$

And consequently, our contributed iterative method is presented in the following three-step view

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \frac{f^2(x_n) + f^2(y_n)}{f'(x_n)[f(x_n) - f(y_n)]}, \\ x_{n+1} = z_n - \frac{f(z_n)}{f[z_n, y_n] + f[z_n, x_n, x_n](z_n - y_n)}. \end{cases} \quad (2.6)$$

In the next section, we will show that our new proposed method (2.6) has seventh-order of convergence. Per one iteration, method (2.6) requires three evaluations of the function and one evaluation of its first derivative, so its efficiency index attains $\sqrt[4]{7} \approx 1.627$ and this is the main motivation of our paper.

3 Convergence Analysis

Theorem 3.1 Let $x^* \in D$ be a simple zero of a sufficiently differentiable function $f : D \rightarrow R$ for an open interval D . Then, the new three-step method that is defined by eq. (2.6) has the seventh-order convergence and satisfies the following error equation,

$$e_{n+1} = 4c_2c_3(4c_2c_3 - c_4 - 4c_2^3)e_n^7 + O(e_n^8),$$

where $e_n = x_n - x^*$, $c_k = (1/k!)f^{(k)}(x^*)/f'(x^*)$.

Proof Let $e_n = x_n - x^*$ and x^* be a simple zero of f . Using Taylor expansion and taking into account $f(x^*)=0$, we have

$$f(x_n) = f'(x^*) \left[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + O(e_n^6) \right] \quad (3.1)$$

and

$$f'(x_n) = f'(x^*) \left[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4 + 6c_6e_n^5 + O(e_n^6) \right]. \quad (3.2)$$

Dividing eq. (3.1) by eq. (3.2) gives us

$$\begin{aligned} \frac{f(x_n)}{f'(x_n)} &= e_n - c_2e_n^2 + 2(c_2^2 - c_3)e_n^3 + (7c_2c_3 - 4c_2^3 - 3c_4)e_n^4 \\ &\quad + 2(4c_2^4 - 10c_2^2c_3 + 3c_3^2 + 5c_2c_4 - 2c_5)e_n^5 + O(e_n^6), \end{aligned} \quad (3.3)$$

and hence

$$\begin{aligned} y_n &= x^* + c_2 e_n^2 - 2(c_2^2 - c_3)e_n^3 - (7c_2c_3 - 3c_4 - 4c_2^3)e_n^4 \\ &\quad - 2(4c_2^4 - 10c_2^2c_3 + 3c_3^2 + 5c_2c_4 - 2c_5)e_n^5 + O(e_n^6). \end{aligned} \quad (3.4)$$

Again, expanding $f(y_n)$ about x^* and then using eq.(3.4), we have

$$\begin{aligned} f(y_n) &= f'(x^*) \left[c_2 e_n^2 - 2(c_2^2 - c_3)e_n^3 - (7c_2c_3 - 3c_4 - 5c_2^3)e_n^4 \right. \\ &\quad \left. - 2(6c_2^4 - 12c_2^2c_3 + 3c_3^2 + 5c_2c_4 - 2c_5)e_n^5 + O(e_n^6) \right]. \end{aligned} \quad (3.5)$$

After an elementary calculation from eqs. (3.1) and (3.5), we obtain

$$\begin{aligned} f(x_n) - f(y_n) &= f'(x^*) \left[e_n + (2c_2^2 - c_3)e_n^3 + (7c_2c_3 - 5c_2^3 - 2c_4)e_n^4 \right. \\ &\quad \left. + 2(6c_2^4 - 12c_2^2c_3 + 3c_3^2 + 5c_2c_4 - 1.5c_5)e_n^5 + O(e_n^6) \right] \end{aligned} \quad (3.6)$$

and

$$\begin{aligned} f^2(x_n) + f^2(y_n) &= f'^2(x^*) \left[e_n^2 + 2c_2e_n^3 + 2(c_3 + c_2^2)e_n^4 + 2(c_4 + 3c_2c_3 - 2c_2^3)e_n^5 \right. \\ &\quad \left. + (14c_2^4 - 22c_2^2c_3 + 5c_3^2 + 8c_2c_4 + 2c_5)e_n^6 + O(e_n^7) \right]. \end{aligned} \quad (3.7)$$

From eqs. (3.2), (3.6) and (3.7), we can obtain

$$\begin{aligned} \frac{f^2(x_n) + f^2(y_n)}{f'(x_n)[f(x_n) - f(y_n)]} &= e_n + 2(4c_2c_3 - c_4 - 4c_2^3)e_n^4 \\ &\quad + (40c_2^4 - 58c_2^2c_3 + 8c_3^2 + 16c_2c_4 - 3c_5)e_n^5 + O(e_n^6). \end{aligned} \quad (3.8)$$

From eq. (3.8), we obtain

$$z_n - x^* = -2(4c_2c_3 - c_4 - 4c_2^3)e_n^4 - (40c_2^4 - 58c_2^2c_3 + 8c_3^2 + 16c_2c_4 - 3c_5)e_n^5 + O(e_n^6) \quad (3.9)$$

and

$$f(z_n) = f'(x^*) \left[\alpha_n + O(\alpha_n^2) \right], \quad \alpha_n = z_n - x^*. \quad (3.10)$$

Thus,

$$\frac{f(z_n)}{f[z_n, y_n] + f[z_n, x_n, x_n](z_n - y_n)} = \alpha_n \left[1 + 2c_2c_3e_n^3 + O(e_n^4) \right]. \quad (3.11)$$

Hence, we obtain the following error equation

$$\begin{aligned} e_{n+1} &= \alpha_n - \frac{f(z_n)}{f[z_n, y_n] + f[z_n, x_n, x_n](z_n - y_n)} \\ &= 4c_2c_3(4c_2c_3 - c_4 - 4c_2^3)e_n^7 + O(e_n^8). \end{aligned} \quad (3.12)$$

This means that the method defined by eq. (2.6) has the seventh-order convergence.

Remark 1 The efficiency index of our scheme defined by eq. (2.6) is $\sqrt[4]{7} \approx 1.627$ which is bigger than $\sqrt{2} \approx 1.414$ of Newton's method. Although the new scheme has lower efficiency index in contrast with $\sqrt[4]{8} \approx 1.682$ of optimal eighth-order methods in [6–7], its convergence radius and accuracy for not so close starting points are better than the accuracy and convergence radii of the optimal eighth-order methods, see Section 4.

Table 1: Comparison of various methods in terms of needed iterations to obtain the root x^*

f	Guess	(4.1)	(4.2)	(4.3)	(4.4)	(4.5)	(2.6)
f_1	-0.3	7 (28)	4 (16)	4 (16)	Div	Div	4 (16)
	-1.6	Div	4 (16)	4 (16)	Div	4 (16)	4 (16)
	-4.0	Div	4 (16)	4 (16)	Div	Div	4 (16)
f_2	0.1	5 (20)	3 (12)	3 (12)	3 (15)	3 (12)	3 (12)
	1.4	8 (32)	4 (16)	5 (20)	4 (20)	4 (16)	4 (16)
f_3	0.42	6 (24)	4 (16)	4 (16)	4 (20)	3 (12)	3 (12)
	-1.42	7 (28)	3 (12)	3 (12)	4 (20)	3 (12)	3 (12)
f_4	-0.5	7 (28)	20 (80)	19 (76)	Div	4 (16)	3 (12)
	-1.9	48 (192)	4 (16)	4 (16)	4 (20)	4 (16)	4 (16)
f_5	-0.5	9 (36)	12 (48)	19 (76)	Div	4 (16)	4 (16)
	3.0	Div	4 (16)	4 (16)	4 (20)	4 (16)	4 (16)

4 Numerical Examples

To show the reliability of the new method (2.6), we compare the results with Zhang's seventh-order methods [5] and the eighth-order methods in [6–7]. Zhang's seventh-order methods are expressed as

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - \frac{1}{1 - 2\alpha + \beta\alpha^2 + \gamma\alpha^3} \frac{f(y_n)}{f'(x_n)}, \\ x_{n+1} = z_n - \frac{f(z_n)}{\lambda}, \end{cases} \quad (4.1)$$

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - (1 + 2\alpha) \frac{f(y_n)}{f'(x_n)}, \\ x_{n+1} = z_n - \frac{f(z_n)}{\lambda}, \end{cases} \quad (4.2)$$

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - (1 + \alpha)^2 \frac{f(y_n)}{f'(x_n)}, \\ x_{n+1} = z_n - \frac{f(z_n)}{\lambda}, \end{cases} \quad (4.3)$$

where $\alpha = f(y_n)/f(x_n)$, $\lambda = f[z_n, y_n] + f[z_n, x_n, x_n](z_n - y_n)$, $\beta, \gamma \in R$. The optimal efficient eighth-order method of Thukral and Petkovic [6] and the optimal novel eighth-order method of Neta and Petkovic [7] are expressed as

Table 2: Comparison of various iterative methods

	(4.1)	(4.2)	(4.3)	(4.4)	(4.5)	(2.6)
δ_1	Div	0.47E-042	0.18E-016	Div	Div	0.16E-084
$ f_1 $	Div	0.00E-000	0.41E-118	Div	Div	0.20E-198
<i>COC</i>	Div	5.97	5.74	Div	Div	6.99
δ_2	0.57E-021	0.24E-016	0.37E-032	0.18E-063	0.88E-058	0.86E-068
$ f_2 $	0.81E-043	0.63E-118	0.26E-198	0.13E-197	0.00E-000	0.3E-199
<i>COC</i>	2.00	6.64	6.50	7.91	7.43	6.64
δ_3	0.17E-018	0.79E-065	0.43E-100	0.17E-055	0.60E-022	0.17E-075
$ f_3 $	0.86E-037	0.00E-000	0.00E-000	0.00E-000	0.86E-177	0.38E-110
<i>COC</i>	2.00	7.00	7.00	7.91	8.56	8.26
δ_4	0.79E-015	0.46E-034	0.45E-015	Div	0.85E-029	0.35E-030
$ f_4 $	0.30E-029	0.00E-000	0.00E-000	Div	0.00E-000	0.00E-000
<i>COC</i>	1.00	6.33	6.50	Div	7.31	6.58

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - \frac{f(x_n) + bf(y_n)}{f(x_n) + (b-2)f(y_n)} \cdot \frac{f(y_n)}{f'(x_n)}, \\ x_{n+1} = z_n - \left[\varphi(n) + \frac{f(z_n)}{f(y_n) - af(z_n)} + \frac{4f(z_n)}{f(x_n)} \right] \cdot \frac{f(z_n)}{f'(x_n)} \end{cases} \quad (4.4)$$

and

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - \frac{f(x_n) + tf(y_n)}{f(x_n) + (t-2)f(y_n)} \cdot \frac{f(y_n)}{f'(x_n)}, \\ x_{n+1} = y_n + c[f(x_n)]^2 - d[f(x_n)]^3, \end{cases} \quad (4.5)$$

where where $a, b, t \in R$ and

$$\begin{cases} \varphi(n) = 1 + 2\kappa_n + (5 - 2b)(\kappa_n)^2 + 2(6 - 6b + b^2)(\kappa_n)^3, \quad \kappa_n = f(y_n)/f(x_n), \\ c = \frac{1}{f(y_n) - f(x_n)} \left(\frac{1}{f[y_n, x_n]} - \frac{1}{f'(x_n)} \right) - d(f(y_n) - f(x_n)), \\ d = \frac{1}{f(y_n) - f(x_n)} \cdot \frac{1}{f(y_n) - f(z_n)} \left(\frac{1}{f[y_n, x_n]} - \frac{1}{f'(x_n)} \right) \\ - \frac{1}{f(z_n) - f(x_n)} \cdot \frac{1}{f(y_n) - f(z_n)} \left(\frac{1}{f[z_n, x_n]} - \frac{1}{f'(x_n)} \right). \end{cases} \quad (4.6)$$

Table 3: Comparison the distance of two consecutive approximations δ of various methods to find the root x^* with the same Total Number of Evaluation (TNE=3)

δ	Guess	(4.1)	(4.2)	(4.3)	(4.4)	(4.5)	(2.6)
δ_1	-0.3	0.15E-00	0.16E-05	0.70E-06	Div	Div	0.51E-06
	-1.6	0.14E+03	0.11E-02	0.16E-02	Div	Div	0.34E-06
	-4.0	0.20E+22	0.51E-06	0.50E-02	Div	Div	0.25E-07
δ_2	1.4	0.97E-00	0.77E-02	0.37E-04	0.16E-07	0.10E-06	0.69E-05
	0.1	0.32E-04	0.35E-40	0.22E-40	0.58E-47	0.98E-53	0.12E-40
δ_3	-1.42	0.93E-01	0.74E-15	0.44E-15	0.64E-13	0.32E-16	0.10E-15
δ_4	-0.5	0.26E-01	0.28E-01	0.54E-02	0.33E-01	0.23E-03	0.54E-03
	-1.9	0.31E-01	0.12E-04	0.45E-04	0.13E-05	0.15E-07	0.46E-04
δ_5	-0.5	0.25E+02	0.20E-05	0.20E-09	0.23E-07	0.66E-09	0.16E-12
	3.0	0.68E+05	0.24E-04	0.14E-04	0.15E-04	0.13E-05	0.88E-08

All computations were performed in MATLAB 7.0 with 200 digit floating arithmetic (VPA=200). When the stopping criterion

$$|x_{n+1} - x_n| + |f(x_{n+1})| < \varepsilon \quad (4.7)$$

is satisfied, x_{n+1} is taken as an approximate of the exact root x^* . We use the following functions:

- $f_1(x) = \sqrt{x^4 + 8} \sin\left(\frac{\pi}{x^2 + 2}\right) + \frac{x^3}{x^4 + 1} - \sqrt{6} + \frac{8}{17}$, $x^* = -2$,
- $f_2(x) = \sin x - 0.5$, $x^* = 0.523598775598299$,
- $f_3(x) = e^x - 4x^2 = 0$, $x^* = 0.7148059123627778$,
- $f_4(x) = \left(\sin x - \frac{\sqrt{2}}{2}\right)^2(x + 1)$, $x^* = -1$,
- $f_5(x) = \cos x - x$, $x^* = 0.739085133215161$.

Displayed in Table 1 is the number of iterations (n) and the number of function evaluations (NFE) required such that the stopping criteria satisfied, where ε is taken to be 10^{-15} . Moreover, displayed in Table 2 is the distance of two consecutive approximations $\delta = |x_{n+1} - x_n|$, the value of $|f(x_{n+1})|$ after the required iterations and the computational order of convergence (COC) which can be approximated using the formula [8]

$$COC = \frac{\ln |(x_{n+1} - x_n)(x_n - x_{n-1})^{-1}|}{\ln |(x_n - x_{n-1})(x_{n-1} - x_{n-2})^{-1}|}. \quad (4.8)$$

The results in Table 1 and Table 2 show that for most of the functions we tested, the present method improves the computational efficiency of Newton's method, the modified Jarratt's method defined by eq. (1.2), where the initial guesses of f_1, f_2, f_3, f_4 in Table 2 are -4.0, 1.4, -0.42 and -1.9, respectively. Note that for the eighth-order methods defined by eqs. (4.4)–(4.5), where $a=b=t=0$, we should pull the attention toward this, which they have a very low convergence radii and that is why in most of the case; they turn out to divergence when the starting points are in the vicinity of the zero but not so close. The applications of

such eighth-order methods as eqs. (4.4)–(4.5) whose convergence radii, are low, are indeed restricted in practice.

Table 4: Comparison the values of $|f(x_{n+1})|$ of various methods to find the root x^* with the same Total Number of Evaluation (TNE=3)

$ f $	Guess	(4.1)	(4.2)	(4.3)	(4.4)	(4.5)	(2.6)
$ f_1 $	-0.3	0.49E-02	0.22E-41	0.67E-44	Div	Div	0.59E-38
	-1.6	0.12E+1	0.14E-21	0.61E-18	Div	Div	0.37E-46
	-4.0	0.12E+1	0.29E-42	0.58E-17	Div	Div	0.32E-12
$ f_2 $	1.4	0.11E-00	0.21E-16	0.97E-01	0.15E-63	0.76E-58	0.75E-38
	0.1	0.25E-09	0.00E-00	0.00E-00	0.00E-00	0.00E-00	0.00E-00
$ f_3 $	-1.42	0.33E-001	0.10E-105	0.20E-107	0.26E-103	0.12E-130	0.58E-112
$ f_4 $	-0.5	0.10E-02	0.84E-03	0.83E-06	0.58E-01	0.20E-28	0.72E-04
	-1.9	0.38E-02	0.11E-33	0.84E-30	0.18E-45	0.83E-62	0.83E-30
$ f_5 $	-0.5	0.36E+02	0.27E+03	0.15E+06	0.12E+07	0.38E-76	0.56E-92
	3.0	0.68E+05	0.17E-34	0.31E-36	0.13E-40	0.83E-90	0.95E-38

We also provide the Total Number of Evaluations (TNE) for each method to obtain the root up to 15 decimal places in Table 3 and Table 4. As we can see, the contributed method (1.3) is robust and accurate in comparison with other efficient schemes. By comparisons with eqs. (4.4)–(4.5), we could claim that the method compete the optimal eighth-order schemes in [6–7] while its computational complexity is less and its convergence radius is bigger too. Of course, if the initial guesses be enough close (very close) to the sought zeros, then the optimal eighth-order methods (4.4)–(4.5) will perform better than (1.3).

Remark 2 Let r be the total operations (including additions, subtractions, divisions, multiplications and so on) of an iterative method per iteration, then the (extended) computational index (also known as extended operational index) is defined by $\sqrt[p]{r}$, where p is the order of convergence. Now, we can compare the computational index of some well known high-order methods with our scheme. The computational index of our method is $\sqrt[25]{6} \approx 1.074$ which is bigger than $\sqrt[39]{8} \approx 1.054$ of (4.4) and $\sqrt[58]{8} \approx 1.036$ of (4.5).

5 Conclusions

In this work we present an efficient Newton-type method with eighth-order convergence by combining previous methods for solving nonlinear equations. Analysis of efficiency shows that the new algorithm is more efficient and it performs better than classical Newton's method and some other methods.

References

- [1] Chun C. Iterative methods improving Newton's methods by the decomposition method[J]. Comput. Math. Appl., 2005, 50(10-12): 1559–1568.

- [2] Chun C. Construction of Newton-like iteration methods for solving non-linear equations[J]. Numer. Math., 2006, 104(3): 297–315.
- [3] Kou Jisheng, Wang Xiuhua, Li Yitian, Some eight-order root-finding three-step methods[J]. Commun. Nonlinear Sci. Numer. Simulat., 2010, 15(3): 536–544.
- [4] Al-Subaihi I. A. A ninth-order iterative method free from second derivative for solving nonlinear equations[J]. Int. J. Math. Analysis, 2011, 5(47): 2337–2347.
- [5] Zhang Y X, Ding H F, He W S, Yang X Y. A new family of methods for nonlinear equations[J]. ICICA 2010, LNCS 6377, 2010: 387–394.
- [6] Thukral R, Petkovic M S. A family of three-point methods of optimal order for solving nonlinear equations[J]. J. Comput. Appl. Math., 2010, 233(9): 2278–2284.
- [7] Neta B, Petkovic M S. Construction of optimal order nonlinear solvers using inverse interpolation[J]. Appl. Math. Comput., 2010, 217(6): 2448–2455.
- [8] Al-Subaihi I A. A ninth-order iterative method free from second derivative for solving nonlinear equations[J]. Int. J. Math. Anal., 2011, 5(47): 2337–2347.

求解非线性方程七阶收敛的牛顿迭代修正格式

王晓峰^{1,2}, 石东洋¹

(1. 郑州大学数学与统计学院, 河南 郑州 450001)

(2. 河南科技学院数学科学学院, 河南 新乡 453003)

摘要: 本文研究了非线性方程求解的问题. 利用泰勒公式和耦合方法, 获得了一种求解非线性方程的加速收敛的七阶迭代改进格式, 该格式不需要计算高阶导数, 且具有更大的收敛半径, 大大提高了计算效率.

关键词: 迭代法; 非线性方程; 扩展指数; 效率指数; 收敛半径

MR(2010)主题分类号: 65H04; 65H05 中图分类号: O241.7