Applied Mathematics and Computation 224 (2013) 358–361

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

On the development of iterative methods for multiple roots

Beny Neta [a,*], Changbum Chun [b], Melvin Scott [c]

[a] Naval Postgraduate School, Department of Applied Mathematics, Monterey, CA 93943, USA
[b] Department of Mathematics, Sungkyunkwan University, Suwon 440-746, Republic of Korea
[c] 494 Carlton Court, Ocean Isle Beach, NC 28469, USA

ARTICLE INFO

Keywords:
Iterative methods
Order of convergence
Multiple roots

ABSTRACT

There are very few optimal fourth order methods for solving nonlinear algebraic equations having roots of multiplicity m. Here we compare 4 such methods, two of which require the evaluation of the $(m-1)^{st}$ root. We will show that such computation does not affect the overall cost of the method.

Published by Elsevier Inc.

1. Introduction

There is a vast literature on the solution of nonlinear equations and nonlinear systems, see for example Ostrowski [1], Traub [2], Neta [3] and the recent book by Petković et al. [4] and references therein. Most of the algorithms are for finding a simple root of a nonlinear equation $f(x)=0$, i.e. for a root α we have $f(\alpha)=0$ and $f'(\alpha)\neq 0$. In this paper we are interested in the case that α is a root of multiplicity $m>1$. Clearly, one can use the quotient $f(x)/f'(x)$ which has a simple root where $f(x)$ has a multiple root. Such an idea will not require a knowledge of the multiplicity, but on the other hand will require higher derivatives. For example, Newton's method for the function $F(x)=f(x)/f'(x)$ will be

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

If we define the efficiency index of a method of order, p as

$$I=p^{1/d}, \tag{2}$$

where d is the number of function- (and derivative-) evaluation per step then this method has an efficiency of $2^{1/3}=1.2599$ instead of $\sqrt{2}=1.4142$ for Newton's method for simple roots.

There are very few methods for multiple roots when the multiplicity is known. The first one is due to Schröder [5] and it is also referred to as modified Newton,

$$x_{n+1}=x_n-mu_n, \tag{3}$$

where

$$u_n=\frac{f(x_n)}{f'(x_n)}. \tag{4}$$

The method is based on Newton's method for the function $G(x)=\sqrt[m]{f(x)}$ which obviously has a simple root at α, the multiple root with multiplicity m of $f(x)$.

* Corresponding author.
E-mail addresses: bneta@nps.edu (B. Neta), cbchun@skku.edu (C. Chun), MScott8223@atmc.net (M. Scott).

0096-3003/$ - see front matter Published by Elsevier Inc.
http://dx.doi.org/10.1016/j.amc.2013.08.077

Another method based on the same G is Laguerre's-like method

$$x_{n+1} = x_n - \frac{\lambda u_n}{1 + sgn(\lambda - m)\sqrt{\left(\frac{\lambda - m}{m}\right)\left[(\lambda - 1) - \lambda u_n \frac{f''(x_n)}{f'(x_n)}\right]}} \tag{5}$$

where λ ($\neq 0, m$) is a real parameter. When $f(x)$ is a polynomial of degree n, this method with $\lambda = n$ is the ordinary Laguerre method for multiple roots, see Bodewig [6] and Neta and Chun [7]. This family of merthods converges cubically.

We now list optimal fourth order methods for multiple roots. The first paper is by Li et al. [8]. Their method is a special case of one of the families found later by Li et al. [9].

Li et al. [9] have developed six fourth order methods based on the results of Neta and Johnson [10] and Neta [11]. Here we list the two optimal fourth order.

- LCN5

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2}u_n, \\ x_{n+1} &= x_n - a_3\frac{f(x_n)}{f'(y_n)} - \frac{f(x_n)}{b_1 f'(x_n) + b_2 f'(y_n)}, \end{aligned} \tag{6}$$

where

$$\begin{aligned} a_3 &= -\frac{1}{2}\frac{\left(\frac{m}{m+2}\right)^m m(m-2)(m+2)^3}{m^3-4m+8}, \\ b_1 &= -\frac{(m^3-4m+8)^2}{m(m^4+4m^3-4m^2-16m+16)(m^2+2m-4)}, \\ b_2 &= \frac{m^2(m^3-4m+8)}{\left(\frac{m}{m+2}\right)^m (m^4+4m^3-4m^2-16m+16)(m^2+2m-4)}. \end{aligned}$$

- LCN6

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2}u_n, \\ x_{n+1} &= x_n - a_3\frac{f(x_n)}{f'(x_n)} - \frac{f(x_n)}{b_1 f'(x_n) + b_2 f'(y_n)}, \end{aligned} \tag{7}$$

where

$$a_3 = -\frac{1}{2}m(m-2),$$

$$b_1 = -\frac{1}{m}, \quad b_2 = \frac{1}{m\left(\frac{m}{m+2}\right)^m}.$$

Zhou et al. [14] have also developed fourth-order optimal methods for multiple roots but they will not be included in the comparison given here. We now give the optimal methods due to Liu and Zhou [12]. These methods require the computation of the $\sqrt[(m-1)]{\frac{f'(y_n)}{f'(x_n)}}$.

Two methods from the family developed by Liu and Zhou [12]

$$\begin{aligned} y_n &= x_n - mu_n, \\ x_{n+1} &= x_n - mH(w_n)\frac{f(x_n)}{f'(x_n)}, \end{aligned} \tag{8}$$

where

$$w_n = \sqrt[(m-1)]{\frac{f'(y_n)}{f'(x_n)}} \tag{9}$$

and $H(0) = 0, \quad H'(0) = 1, \quad H''(0) = \frac{4m}{m-1}$.

The two members given there are.

- LZ11

Table 1
The functions used for comparison.

Function	x_0	α	m
x^4-2x^2+1	1.6	1.	2
x^6-2x^3+1	1.6	1.	2
$(\sin^2 x-x^2+1)^2$	4.5	1.40449	2
$(\sin^2 x-x^2+1)^2$	2.5	1.40449	2
$(e^x+x-20)^2$	3.0	2.84244	2
$(e^x+x-20)^2$	5.5	2.84244	2
$x^{15}-3x^{10}+3x^5-1$	3.5	1.	3
$x^{18}-3x^{12}+3x^6-1$	1.6	1.	3
$(\cos x-x)^3$	1.5	.739085	3
$(\cos x-x)^3$	2.5	.739085	3
$(x^3+4x^2-10)^3$	3.0	1.36523	3
$(x^3+4x^2-10)^3$	−.4	1.36523	3
$x^{28}-4x^{21}+6x^{14}-4x^7+1$	4.0	1.	4
$(xe^{x^2}-\sin^2 x+3\cos x+5)^4$	3.5	−1.20765	4
$(xe^{x^2}-\sin^2 x+3\cos x+5)^4$	2.5	−1.20765	4
$(e^{x^2+7x-30}-1)^4$	3.25	3	4
$(e^{x^2+7x-30}-1)^4$	5.	3	4
$(\ln(x)+\sqrt{x}-5)^4$	.5	8.30943	4
$(\ln(x)+\sqrt{x}-5)^4$	10.	8.30943	4
$(x^4-2x^2+1)^2$	1.6	1.	4
$(e^x+x-20)^4$	3.0	2.84244	4
$x^{20}-5x^{16}+10x^{12}-10x^8+5x^4-1$	1.6	1.	5
$(x^2-e^x-3x+2)^5$	1.8	.25753	5
$(x^2-e^x-3x+2)^5$	2.0	.25753	5
$(e^x+x-20)^5$	3.0	2.84244	5
$(\cos x-x)^5$	1.5	.739085	5
$(e^x+x-20)^6$	3.0	2.84244	6
$(x^4-2x^2+1)^3$	1.6	1.	6
$(x^{18}-3x^{12}+3x^6-1)^2$	1.6	1.	6
$(\cos x-x)^6$	1.5	.739085	6
$(x^3+4x^2-10)^6$	3.0	1.36523	6

Table 2
Timing for the 4 optimal methods to run all 31 examples.

LCN5	4.39 sec
LCN6	5.05 sec
LZ11	5.17 sec
LZ12	16.03 sec

$$y_n = x_n - m\frac{f(x_n)}{f'(x_n)},$$
$$x_{n+1} = y_n - m\left(w_n + \frac{2m}{m-1}w_n^2\right)\frac{f(x_n)}{f'(x_n)}, \tag{10}$$

- LZ12

$$y_n = x_n - m\frac{f(x_n)}{f'(x_n)},$$
$$x_{n+1} = y_n + \frac{(m-1)w_n}{1-m+2mw_n}\frac{f(x_n)}{f'(x_n)}. \tag{11}$$

The last two methods (denoted LZ11 and LZ12) are the only ones known to the authors where the root of the function is required at each step. It may be that this will increase the cost of the method, therefore in the next section we ran these two

methods for multiple roots with multiplicity $m > 2$ in order to see the effect of having to evaluate $\sqrt[(m-1)]{\frac{f'(y_n)}{f'(x_n)}}$. We compare that to other optimal fourth order methods listed above (namely, LCN5 and LCN6).

In the next section we give timing comparison for optimal fourth order methods that require $\sqrt[(m-1)]{\frac{f'(y_n)}{f'(x_n)}}$ and those that do not. We close with concluding remarks.

2. Comparison of optimal fourth order methods

Kung and Traub [13] conjectured that multipoint iterative methods without memory, requiring $n+1$ function-evaluations per iteration, have order of convergence at most 2^n. Multipoint methods that satisfy the Kung-Traub conjecture are called optimal methods. Their efficiency index is $2^{n/(n+1)}$. Of all the methods listed above we have only 4 optimal methods of order 4, namely (6), (7), (10), and (11). We refer to these methods as LCN5, LCN6, LZ11 and LZ12, respectively. We have ran these 4 methods on 31 cases having roots of multiplicty $m = 2, 3, 4, 5$. Table 1 gives the functions, the initial guess, the root and its multiplicity.

We have ran Maple using Digits:=128, the tolerance for convergence is 10^{-25} and no more than 1000 iterations. The timing required for all examples for each method is given in Table 2.

It is clear that LZ12 requires more than three times as much CPU time than the other three methods use. Since LZ11 does not require more CPU time than LCN5 and LCN6, we conclude that the reason for the extra cost is due to a slow convergence of LZ12 and not that the $(m-1)^{st}$ root is costly. As suggested by the referees, we have looked at the reason for the time consumption of LZ12 and found that the method requires more iterations to converge. In a follow-on manuscript we will compare the basins of attraction which hopefully will show that unless one starts very close to the root, LZ12 will require more iterations.

Conclusion

The computation of $\sqrt[(m-1)]{\frac{f'(y_n)}{f'(x_n)}}$ does not significantly increase the CPU time required for convergence.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0007160).

References

[1] A.M. Ostrowski, Solution of Equations in Euclidean and Banach Space, Academic Press, New York, 1973.
[2] J.F. Traub, Iterative Methods For The Solution of Equations, Chelsea publishing company, New York, 1977.
[3] B. Neta, Numerical methods for the solution of equations, Net-A-Sof, California, 1983.
[4] M.S. Petković, B. Neta, L.D. Petković, J. Džunić, Multipoint Methods for Solving Nonlinear Equations, Elsevier, Waltham, MA, 2013.
[5] E. Schröder, Über unendlich viele Algorithmen zur Auflösung der Gleichungen, Math. Ann. 2 (1870) 317–365.
[6] E. Bodewig, Sur la méthode Laguerre pour l'approximation des racines de certaines équations algébriques et sur la critique d'Hermite, Indag. Math. 8 (1946) 570–580.
[7] B. Neta, C. Chun, On a family of Laguerre methods to find multiple roots of nonlinear equations, Appl. Math. Comput. 219 (2013) 10987–11004.
[8] S. Li, X. Liao, L. Cheng, A new fourth-order iterative method for finding multiple roots of nonlinear equations, Appl. Math. Comput. 215 (2009) 1288–1292.
[9] S.G. Li, L.Z. Cheng, B. Neta, Some fourth-order nonlinear solvers with closed formulae for multiple roots, Comput. Math. Appl. 59 (2010) 126–135.
[10] B. Neta, A.N. Johnson, High order nonlinear solver for multiple roots, Comput. Math. Appl. 55 (2008) 2012–2017.
[11] B. Neta, Extension of Murakami's high order nonlinear solver to multiple roots, Int. J. Comput. Math. 8 (2010) 1023–1031.
[12] B. Liu, X. Zhou, A new family of fourth-order methods for multiple roots of nonlinear equations, Nonlinear Anal.: Model. Control 18 (2013) 143–152.
[13] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iteration, J. ACM 21 (1974) 643–651.
[14] X. Zhou, X. Chen, Y. Song, Constructing higher order methods for multiple roots of nonlinear equations, J. Comput. Appl. Math. 235 (2011) 4199–4206.