CAMWA: 4029 +Model pp. 1–6 (col. fig: NIL)

ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

Computers and Mathematics with Applications xx (xxxx) xxx–xxx

An International Journal computers & mathematics with applications

www.elsevier.com/locate/camwa

High-order nonlinear solver for multiple roots

B. Neta[a,*], Anthony N. Johnson[b]

[a] *Naval Postgraduate School, Department of Applied Mathematics, Monterey, CA 93943, United States*

[b] *United States Military Academy, Department of Mathematical Sciences, West Point, NY 10996, United States*

Abstract

A method of order four for finding multiple zeros of nonlinear functions is developed. The method is based on Jarratt's fifth-order method (for simple roots) and it requires one evaluation of the function and three evaluations of the derivative. The informational efficiency of the method is the same as previously developed schemes of lower order. For the special case of double root, we found a family of fourth-order methods requiring one less derivative. Thus this family is more efficient than all others. All these methods require the knowledge of the multiplicity.

Keywords: Nonlinear equations; High order; Multiple roots; Fixed point

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 references there. Here we develop a high-order fixed point type method to approximate a multiple root. There are several methods for computing a zero ξ of multiplicity m of a nonlinear equation $f(x) = 0$, see Neta [3]. Newton's method is only of first order unless it is modified to gain the second order of convergence, see Rall [4] or Schröder [5]. This modification requires a knowledge of the multiplicity. Traub [2] has suggested the use of any method for $f^{(m)}(x)$ or $g(x) = \frac{f(x)}{f'(x)}$. Any such method will require higher derivatives than the corresponding one for simple zeros. Also the first one of those methods requires the knowledge of the multiplicity m. In such a case, there are several other methods developed by Hansen and Patrick [6], Victory and Neta [7], and Dong [8]. Since in general one does not know the multiplicity, Traub [2] suggested a way to approximate it during the iteration.

For example, the quadratically convergent modified Newton's method is

$$x_{n+1} = x_n - m\frac{f_n}{f'_n} \tag{1}$$

and the cubically convergent Halley's method [9] is

$$x_{n+1} = x_n - \frac{f_n}{\frac{m+1}{2m} f'_n - \frac{f_n f''_n}{2 f'_n}} \tag{2}$$

* Corresponding author.
E-mail address: byneta@gmail.com (B. Neta).

0898-1221/$ - see front matter © 2007 Published by Elsevier Ltd
doi:10.1016/j.camwa.2007.09.001

Please cite this article in press as: B. Neta, A.N. Johnson, High-order nonlinear solver for multiple roots, Computers and Mathematics with Applications (2007), doi:10.1016/j.camwa.2007.09.001

where $f_n^{(i)}$ is short for $f^{(i)}(x_n)$. Another third-order method was developed by Victory and Neta [7] and is based on King's fifth-order method (for simple roots) [10]

$$\begin{aligned} w_n &= x_n - \frac{f_n}{f'_n} \\ x_{n+1} &= w_n - \frac{f(w_n)}{f'_n}\frac{f_n + Af(w_n)}{f_n + Bf(w_n)} \end{aligned} \tag{3}$$

where

$$\begin{aligned} A &= \mu^{2m} - \mu^{m+1} \\ B &= -\frac{\mu^m(m-2)(m-1)+1}{(m-1)^2} \end{aligned} \tag{4}$$

and

$$\mu = \frac{m}{m-1}. \tag{5}$$

Yet two other third-order methods developed by Dong [8], both require the same information and both are based on a family of fourth-order methods (for simple roots) due to Jarratt [11]:

$$x_{n+1} = x_n - u_n - \frac{f(x_n)}{\left(\frac{m}{m-1}\right)^{m+1} f'(x_n - u_n) + \frac{m-m^2-1}{(m-1)^2} f'(x_n)} \tag{6}$$

$$x_{n+1} = x_n - \frac{m}{m+1}u_n - \frac{\frac{m}{m+1}f(x_n)}{\left(1+\frac{1}{m}\right)^m f'\left(x_n - \frac{m}{m+1}u_n\right) - f'(x_n)} \tag{7}$$

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

Our starting point here is Jarratt's method [12] given by the iteration

$$x_{n+1} = x_n - \frac{f(x_n)}{a_1 f'(x_n) + a_2 f'(y_n) + a_3 f'(\eta_n)} \tag{8}$$

where u_n is as above and

$$\begin{aligned} y_n &= x_n - au_n \\ v_n &= \frac{f(x_n)}{f'(y_n)} \\ \eta_n &= x_n - bu_n - cv_n. \end{aligned} \tag{9}$$

Jarratt has shown that this method (for simple roots) is of order 5 [12] if the parameters are chosen as follows

$$a = 1, \qquad b = \frac{1}{8}, \qquad c = \frac{3}{8}, \qquad a_1 = a_2 = \frac{1}{6}, \qquad a_3 = \frac{2}{3}. \tag{10}$$

It requires one function- and three derivative-evaluation per step. Thus the informational efficiency (see [2]) is 1.25. Since Jarratt did not give the asymptotic error constant, we have employed Maple [13] to derive it,

$$\frac{1}{24}A_5 + \frac{1}{2}A_4A_2 - \frac{1}{4}A_3^2 + \frac{1}{8}A_2^2A_3 + A_2^4,$$

where A_i are given by (14) with $m = 1$.

2. New higher order scheme

We would like to find the six parameters a, b, c, a_1, a_2, a_3 so as to maximize the order of convergence to a root ξ of multiplicity m. Let $e_n, \hat{e}_n, \epsilon_n$ be the errors at the nth step, i.e.

$$\begin{aligned} e_n &= x_n - \xi \\ \hat{e}_n &= y_n - \xi \\ \epsilon_n &= \eta_n - \xi. \end{aligned} \tag{11}$$

If we expand $f(x_n)$, and $f'(x_n)$ in Taylor series (truncated after the Nth power, $N > m$) we have

$$f(x_n) = f(x_n - \xi + \xi) = f(\xi + e_n) = \frac{f^{(m)}(\xi)}{m!}\left(e_n^m + \sum_{i=m+1}^{N} A_i e_n^i\right) \tag{12}$$

or

$$f(x_n) = \frac{f^{(m)}(\xi)}{m!} e_n^m \left(1 + \sum_{i=m+1}^{N} B_{i-m} e_n^{i-m}\right) \tag{13}$$

where

$$A_i = \frac{m! f^{(i)}(\xi)}{i! f^{(m)}(\xi)}, \quad i > m$$

$$B_{i-m} = A_i \tag{14}$$

$$f'(x_n) = \frac{f^{(m)}(\xi)}{(m-1)!} e_n^{m-1} \left(1 + \sum_{i=m+1}^{N} \frac{i}{m} B_{i-m} e_n^{i-m}\right). \tag{15}$$

To expand $f'(y_n)$ and $f'(\eta_n)$ we use some symbolic manipulator, such as Maple [13], we find

$$f'(y_n) = \frac{f^{(m)}(\xi)}{(m-1)!} \hat{e}_n^{m-1} \left(1 + \frac{m+1}{m} B_1 \hat{e}_n + \frac{m+2}{m} B_2 \hat{e}_n^2 + \cdots\right) \tag{16}$$

$$\begin{aligned} \hat{e}_n = e_n - a u_n &= \left(1 - \frac{a}{m}\right) e_n + \frac{a}{m^2} B_1 e_n^2 + \left[\frac{2a}{m^2} B_2 - \frac{a(m+1)}{m^3} B_1^2\right] e_n^3 + \cdots \\ &= \frac{1}{2} e_n + \frac{1}{2m} B_1 e_n^2 + \frac{1}{m}\left[B_2 - \frac{m+1}{2m} B_1^2\right] e_n^3 + \cdots \end{aligned} \tag{17}$$

where, for simplicity, we chose

$$a = \frac{m}{2}. \tag{18}$$

Thus

$$f'(y_n) = \frac{f^{(m)}(\xi)}{(m-1)!} e_n^{m-1} (c_0 + c_1 e_n + c_2 e_n^2 + c_3 e_n^3 + \cdots) \tag{19}$$

where

$$\begin{aligned} c_0 &= 2^{1-m} \\ c_1 &= \frac{3m-1}{m} 2^{-m} B_1 \\ c_2 &= \left[\frac{4-2m}{m^2} B_1^2 + \frac{3(3m-2)}{2m} B_2\right] 2^{-m} \\ c_3 &= \left[\frac{25m-21}{4m} B_3 + \frac{m^2-21m+34}{2m^2} B_1 B_2 - \frac{m^3-12m^2-13m+48}{6m^3} B_1^3\right] 2^{-m}. \end{aligned} \tag{20}$$

Please cite this article in press as: B. Neta, A.N. Johnson, High-order nonlinear solver for multiple roots, Computers and Mathematics with Applications (2007), doi:10.1016/j.camwa.2007.09.001

The error in η_n is given by

$$\epsilon_n = e_n - bu_n - cv_n = \lambda e_n + \frac{2b + \hat{c}(m-1)}{2m^2} B_1 e_n^2 + \left[\frac{8b + (5m-6)\hat{c}}{4m^2} B_2 - \frac{4b(m+1) - (3m^2-7)\hat{c}}{4m^3} B_1^2 \right] e_n^3 + \cdots \tag{21}$$

where

$$\hat{c} = 2^{m-1}c, \qquad \lambda = 1 - \frac{b + \hat{c}}{m}. \tag{22}$$

We now expand $f'(\eta_n)$ in terms of e_n

$$f'(\eta_n) = \frac{f^{(m)}(\xi)}{(m-1)!} \epsilon_n^{m-1} \left(1 + \frac{m+1}{m} B_1 \epsilon_n + \frac{m+2}{m} B_2 \epsilon_n^2 + \cdots \right) = \frac{f^{(m)}(\xi)}{(m-1)!} e_n^{m-1} (d_0 + d_1 e_n + d_2 e_n^2 + \cdots) \tag{23}$$

where

$$\begin{aligned} d_0 &= \lambda^{m-1} \\ d_1 &= \frac{\lambda^{m-2} B_1}{m^3} \left\{ (m^2 + b^2)(m+1) - bm(m+3) + (m+1)\hat{c}^2 + \left[2b(m+1) - m\frac{m^2 - 6m - 3}{2} \right] \hat{c} \right\} \\ d_2 &= -\frac{\lambda^{m-3}}{32m^5} B_1^2 [\alpha_1 b + \beta_1 \hat{c} + \gamma_1 \hat{c}^2 + \delta_1 \hat{c}^3] + \frac{\lambda^{m-3}}{m^5} B_2 [\alpha_2 + \beta_2 \hat{c} + \gamma_2 \hat{c}^2 + \delta_2 \hat{c}^3 + \gamma_3 \hat{c}^4] \end{aligned} \tag{24}$$

where

$$\begin{aligned} \alpha_1 &= 16[2m(m+1)b^2 - m^2(m-7)b + 2m^2(m+1)] \\ \beta_1 &= 8m[6b^2(m+1)(m+3) + b(m^3 - 15m^2 - m - 1) - m(m-1)(m^2 - 2m - 7)], \\ \gamma_1 &= 4[8bm^2(m+1) + m(m-1)(m^3 - 6m^2 - 3m - 16) + 4m^2(m-1)] \\ \delta_1 &= 16m^2(m-1) \\ \alpha_2 &= 32[b^4(m+2) - 4b^3m^2 + 2b^2m(m+4)(2m-1) - 2bm^3(m+5) + m^5(m+2)] \\ \beta_2 &= 8[16b^3(m+2) - 48b^2m(m+2) - bm^2(5m^2 - 51m - 98) + m^3(5m^2 - 27m - 26)] \\ \gamma_2 &= 8[24b^2(m+2) - 48bm(m+2) - m^2(5m^2 - 35m - 42)] \\ \delta_2 &= 128[b(m+2) - m(m+1)] \\ \gamma_3 &= 32(m+2). \end{aligned} \tag{25}$$

Now substitute (13), (15), (19) and (23) into (8) and expand the quotient $f_n/(a_1 f'(x_n) + a_2 f'(y_n) + a_3 f'(\eta_n))$ in Taylor series, we get

$$e_{n+1} = e_n - \frac{f_n}{a_1 f'(x_n) + a_2 f'(y_n) + a_3 f'(\eta_n)} = C_1^1 e_n + C_2^1 B_1 e_n^2 + (C_3^1 B_1^2 + C_3^2 B_2) e_n^3 + (C_4^1 B_1^3 + C_4^2 B_1 B_2 + C_4^3 B_3) e_n^4 + \cdots \tag{26}$$

where the coefficients C_i^j depend on the parameters b, c, a_1, a_2, a_3. These 5 parameters can be used to annihilate the coefficients of e_n, e_n^2, e_n^3 and one of the terms in e_n^4. Thus the method is of order $p = 4$. Actually, except for $m = 2$, we used $b = a = m/2$ and thus we have only 4 parameters at our disposal. This is sufficient to obtain fourth-order methods.

Table 1
Results for Example 2

n	x	f	x	f
0	0.8	0.1296	0.6	0.4096
1	1.00074058	0.21954564(−5)	1.02772277	0.31600247(−2)
2			1.00000014	0.750396(−13)

Because of the complexity of the above equations, we have listed the parameters for $m = 2, 3, 4, 5$ and 6. All these methods are of fourth order.

m	2	2	3	4	5	6
a	1	$\frac{4}{3}$	$\frac{3}{2}$	2	$\frac{5}{2}$	3
b	free	free	free	2	$\frac{5}{2}$	3
c	free	$\frac{1-b}{3}$	$\frac{3}{5} - \frac{b}{4}$	0.06478279184	0.0217372041	0.0082119760
a_1	$-\frac{1}{2}$	$\frac{1-2b}{2}$	$\frac{25}{108}b - \frac{43}{72}$	−0.4374579865	−0.4303454005	−0.3681491853
a_2	2	$3(b-1)$	$4 - \frac{25}{72}b$	7.90412890309	18.8154365391	39.6876826792
a_3	0	2	$-\frac{125}{72}$	−5.9128176652	−15.8940830499	−35.6993794378
r_1	$-\frac{1}{2}$	$\frac{2}{9}b - \frac{13}{18}$	$\frac{5b}{1296} - \frac{37}{108}$	−0.2362609294	−0.1647909926	−0.1201790024
r_2	$\frac{3}{8}$	$\frac{7}{8} - \frac{b}{2}$	$\frac{25}{81} - \frac{5b}{972}$	0.1546752539	0.1013867224	0.07303104907
r_3	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{2}{25}$	0.08352683535	0.06967247928	0.05702535018

The error is given by

$$e_{n+1} = (r_1 B_1 B_2 + r_2 B_1^3 + r_3 B_3) e_n^4 \tag{27}$$

where r_1, r_2, and r_3 are given in the above table for each m. For $m = 3$, we can choose the free parameter b to equal $a = 3/2$.

To summarize, we managed to obtain a fourth-order method requiring one function- and three derivative-evaluation per step. The informational efficiency of these methods is 1, as all the above mentioned methods for multiple roots. The efficiency index is 1.4142 which is lower than the third-order methods. In the case $m = 2$ we found a method that will require only two derivative-evaluations ($a_3 = 0$) and thus the informational efficiency is $4/3$ and the efficiency index is 1.5874. We could not find such efficient methods for higher m.

3. Numerical experiments

In our first example we took a quadratic polynomial having a double root at $\xi = 1$

$$f(x) = x^2 - 2x + 1. \tag{28}$$

Here we started with $x_0 = 0$ and the convergence is achieved in 1 iteration. In the second example we took a polynomial having two double roots at $\xi = \pm 1$

$$f(x) = x^4 - 2x^2 + 1. \tag{29}$$

Starting at $x_0 = 0.8$, our method converged in 1 iteration. When we start at $x_0 = 0.6$, our method required 2 iterations. The results are given in Table 1.

Similar results were obtained when starting at $x_0 = -0.8$ and $x = -0.6$ to converge to $\xi = -1$.

The next example is a polynomial with triple root at $\xi = 1$

$$f(x) = x^5 - 8x^4 + 24x^3 - 34x^2 + 23x - 6. \tag{30}$$

Please cite this article in press as: B. Neta, A.N. Johnson, High-order nonlinear solver for multiple roots, Computers and Mathematics with Applications (2007), doi:10.1016/j.camwa.2007.09.001

Table 2
Results for Example 3

n	x	f
0	0	−6.
1	0.95239072	−0.23148417(−3)
2	0.99999683	−0.63(−16)

Table 3
Results for Example 4

n	x	f	x	f
0	0.1	0.11051709(−1)	0.2	0.48856110(−1)
1	0.12654311(−4)	0.16013361(−9)	0.17709827(−3)	0.31369352(−7)
2	0.3739(−20)	0	0.14341725(−15)	0

Table 4
Results for Example 5

n	x	f
0	0	19.
1	1.46056319	9.725126111
2	1.00101187	0.368806435(−4)
3	1.	0.

The iteration starts with $x_0 = 0$ and the results are summarized in Table 2. Another example with a double root at $\xi = 0$ is

$$f(x) = x^2 e^x. \tag{31}$$

Starting at $x_0 = 0.1$ our method converged in 1 iteration, but when we start at $x_0 = 0.2$, our scheme converged in 1 iteration. The results are given in Table 3. The last example having a double root at $\xi = 1$ is

$$f(x) = 3x^4 + 8x^3 - 6x^2 - 24x + 19. \tag{32}$$

Now we started with $x_0 = 0$ and the results are summarized is Table 4.

References

[1] A.M. Ostrowski, Solutions of Equations and System of Equations, Academic Press, New York, 1960.
[2] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice Hall, New Jersey, 1964.
[3] B. Neta, Numerical Methods for the Solution of Equations, Net-A-Sof, California, 1983.
[4] L.B. Rall, Convergence of the Newton process to multiple solutions, Numer. Math. 9 (1966) 23–37.
[5] E. Schröder, Über unendlich viele Algorithmen zur Auflösung der Gleichungen, Math. Ann. 2 (1870) 317–365.
[6] E. Hansen, M. Patrick, A family of root finding methods, Numer. Math. 27 (1977) 257–269.
[7] H.D. Victory, B. Neta, A higher order method for multiple zeros of nonlinear functions, Int. J. Comput. Math. 12 (1983) 329–335.
[8] C. Dong, A family of multipoint iterative functions for finding multiple roots of equations, Int. J. Comput. Math. 21 (1987) 363–367.
[9] E. Halley, A new, exact and easy method of finding the roots of equations generally and that without any previous reduction, Phil. Trans. R. Soc. London 18 (1694) 136–148.
[10] R.F. King, A family of fourth order methods for nonlinear equations, SIAM J. Numer. Anal. 10 (1973) 876–879.
[11] P. Jarratt, Some fourth order multipoint methods for solving equations, Math. Comp. 20 (1966) 434–437.
[12] P. Jarratt, Multipoint iterative methods for solving certain equations, Comput. J. 8 (1966) 398–400.
[13] D. Redfern, The Maple Handbook, Springer-Verlag, New York, 1994.

Please cite this article in press as: B. Neta, A.N. Johnson, High-order nonlinear solver for multiple roots, Computers and Mathematics with Applications (2007), doi:10.1016/j.camwa.2007.09.001