International Electronic Journal of Pure and Applied Mathematics
Volume 3 No. 3 2011, 245-252

ON A CLASS OF FIFTEENTH-ORDER ITERATIVE FORMULAS FOR SIMPLE ROOTS

F. Soleymani[1 §], M. Sharifi[2]
[1,2]Department of Mathematics
Islamic Azad University
Zahedan Branch, Zahedan, IRAN

Abstract: A class of fifteenth-order convergent iterative methods for finding simple roots of nonlinear equations is presented. This new class of without memory methods includes four evaluations of the function and one evaluation of the first derivative per iteration. Therefore, its efficiency index is $15^{1/5} \approx 1.7187$. Convergence analysis confirms the fifteenth-order convergence and also provides the error equations. Some numerical tests are provided to illustrate the efficacy of the new methods of the class.

AMS Subject Classification: 65H05, 41A25
Key Words: nonlinear equations, high-order iterative methods, optimality, order of convergence, efficiency index, simple root

1. Introduction

It is well known that a wide class of problems which arise in various disciplines of mathematical and engineering sciences can be formulated in terms of nonlinear equations of the form $f(x) = 0$. In recent years, much attention have been given to develop and analyze a number of numerical methods for solving such equations. In fact recently, there is a renewed interest in developing higher order algorithms for such problem.

After the locally quadratically Newton's method [11] of optimal order of convergence, a lot of methods have been developed by providing a two-step or three-step cycle for boosting up the order of convergence and using one or two more evaluations of the function or the derivatives per iteration; see e.g. [7, 13, 15]. We remind that for a without memory iterative scheme which is convergent to the simple root of the

Received: March 19, 2011

§Correspondence author

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 3, No. 3 (2011)

single variable nonlinear equation and carries out n evaluations per iteration, then the method is called of optimal convergence order [8] when the convergence order be 2^{n-1}. And consequently, its optimal efficiency index will be $2^{(n-1)/n}$. Now, let us review some high order schemes.

In 1974, Kung and Traub [8] provided a class of n-step derivative-involved methods (by using inverse Hermite interpolation) including n evaluations of the function and one of its first derivative per full iteration to reach the convergence rate 2^n. They also have given a n-step derivative-free family of one parameter (by using the inverse interpolation and consuming $n+1$ evaluations of the function) to again achieve the optimal convergence rate 2^n. Hence, in case of $n = 3$, the schemes reach the order eight with $8^{1/4} \approx 1.6817$ as the efficiency index while for $n = 4$, such techniques achieve the optimal order sixteen with $16^{1/5} \approx 1.7411$ as the efficiency index. In 2009, Bi et al. [1] provided new optimal three-step methods using an adequate approximation for the derivative in the last step and constructing weight functions to reach the optimality. One year later, this procedure was also taken into consideration by Sharma and Sharma in [12], and Geum and Kim in [4] to furnish new optimal three-step without memory schemes and the same efficiency index $8^{1/4} \approx 1.6817$. In 2010, Neta and Petkovic [9] used inverse interpolation for providing optimal three- and four-step methods of order 8 and 16 which achieve the optimal efficiency indices $8^{1/4} \approx 1.6817$ and $16^{1/5} \approx 1.7411$ respectively. We here remark that, Neta in [10] proposed a four-step method using inverse interpolation but did not demonstrate its order of convergence. Newly, Geum and Kim in [5] proved that the order of convergence for Neta's method is fourteen. To see more on this topic; refer to [2, 3, 6, 14].

In this work, we consider a four-step cycle in which the first three steps are any of the optimal eighth-order without memory methods (with three evaluations of the function and one of its first derivative) and the Newton's iteration in the last step, then we use a good approximation for the new-appeared first derivative of the function in the last step to produce a novel class of four-step fifteenth-order methods which possess $15^{1/5} \approx 1.7187$ as their efficiency index. This efficiency index is much better than that of $8^{1/4} \approx 1.6817$ of optimal eighth-order methods and $14^{1/5} \approx 1.6952$ of Neta's fourteenth-order method [10]. Although this class with 1.7187 as the efficiency index is not optimal according to the Kung and Traub conjecture (1974), its computational burden is lower than that of optimal sixteenth-order methods.

2. Development of the New Class

In this section, first we construct a new class of four-step without memory methods with fifteenth-order of convergence by defining a suitable and easy to implement approximation of the derivative in the last step. Due to this, we consider

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - \dfrac{2f(x_n) - f(y_n)}{2f(x_n) - 5f(y_n)} \dfrac{f(y_n)}{f'(x_n)}, \\ w_n = z_n - \dfrac{f(x_n) + 2f(z_n)}{f(x_n)} \dfrac{f(z_n)}{f[z_n, y_n] + f[z_n, x_n, x_n](z_n - y_n)}, \\ x_{n+1} = w_n - \dfrac{f(w_n)}{f'(w_n)}. \end{cases} \tag{1}$$

The first three steps of (1) are the optimal eighth-order method given by Bi et al. in [1]. It is crystal clear that scheme (3) is a sixteenth-order method with six evaluations per iteration. Now, the main question is that "is there any way to keep the order up but reduce the number of evaluations while the method be free from high computational burden". For this cause, by using four past known data (except the known value in the first derivative), a very powerful approximation for $f'(w_n)$ will be obtained. That is, we approximate $f'(w_n)$ in the domain D of the simple zero, by an approximating polynomial of degree three as follows

$$f(t) \approx A(t) = a_0 + a_1(t - x_n) + a_2(t - x_n)^2 + a_3(t - x_n)^3. \tag{2}$$

At this time, the four unknown quantities in (2) should be attained by satisfying in the interpolating conditions $f(x_n) = A(x_n)$, $f(y_n) = A(y_n)$, $f(z_n) = A(z_n)$, and $f(w_n) = A(w_n)$. It is obvious that $a_0 = f(x_n)$. Finally, by solving a system of three linear equations with three unknowns, we can approximate $f'(w_n)$ as follows:

$$f'(w_n) \approx A'(w_n) = a_1 + 2a_2(w_n - x_n) + 3a_3(w_n - x_n)^2 =$$

$$f[x_n, w_n] + (f[y_n, x_n, z_n] - f[y_n, x_n, w_n] - f[z_n, x_n, w_n])(x_n - w_n), \tag{3}$$

where $f[x_n, w_n]$, $f[y_n, x_n, z_n]$, $f[y_n, x_n, w_n]$ and $f[z_n, x_n, w_n]$ are divided differences of the function f. Consequently a novel four-step iterative method can be written in the following way

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - \dfrac{2f(x_n) - f(y_n)}{2f(x_n) - 5f(y_n)} \dfrac{f(y_n)}{f'(x_n)}, \\ w_n = z_n - \dfrac{f(x_n) + 2f(z_n)}{f(x_n)} \dfrac{f(z_n)}{f[z_n, y_n] + f[z_n, x_n, x_n](z_n - y_n)}, \\ x_{n+1} = w_n - \dfrac{f(w_n)}{f[x_n, w_n] + (f[y_n, x_n, z_n] - f[y_n, x_n, w_n] - f[z_n, x_n, w_n])(x_n - w_n)}, \end{cases} \tag{4}$$

in which we have four evaluations of the function and one evaluation of the first derivative per iteration. We shall see that its order of convergence reaches fifteen with only five evaluations per full iteration, which means that the proposed method possesses a good efficiency index, i.e. 1.7187.

Theorem 1. *Let $x^* \in D$ be a simple zero of a sufficiently differentiable function $f : D \subset \mathbb{R} \to \mathbb{R}$ for an open interval D, which includes x_0 as the initial approximation of x^*. Then, the method (4) is of order fifteen and consists of four evaluations of the function and one evaluation of the first derivative.*

Proof. We simply provide the order of convergence by expanding the Taylor's series around the simple root for the function and its first derivative in the nth iterate. Note that $c_k = (\frac{1}{k!})\frac{f^{(k)}(x^*)}{f'(x^*)}$, $k \geq 2$. We seek this problem with the following Mathematica program (Matlab and Maple are also convenient). The following terms are used in the program given below without the index n, $e = x - x^*$, $u = y-x^*$, $v = z-x^*$, $g = w-x^*$, $fx = f(x)$, $fy = f(y)$, $fz = f(z)$, $fw = f(w)$, $dfx = f'(x)$, $dfa = f'(x^*)$.

Mathematica Program:

```
fx[e_]=dfa*(e+c2*e^2+c3*e^3+c4*e^4+c5*e^5+c6*e^6+c7*e^7+c8*e^8
+c9*e^9+c10*e^10+c11*e^11+c12*e^12);dfx[e_]=D[fx[e],e];u=e-
Series[fx[e]/dfx[e],{e,0,15}];fy[u_]=dfa*(u+c2*u^2+c3*u^3+c4*u^4
+c5*u^5);v=u-((2*fx[e]-fy[u])/(2*fx[e]-5*fy[u]))*(fy[u]/dfx[e]);
fz[v_]=dfa*(v+c2*v^2+c3*v^3+c4*v^4);g=v-(fz[v]/(((fz[v]-fy[u])/(v
-u))+((((fz[v]-fx[e])/(v-e))-dfx[e])/(v-e))*(v-u)))*((fx[e]
+2*fz[v])/(fx[e]));fw[g_]=dfa*(g+c2*g^2+c3*g^3+c4*g^4);j=(fx[e]
-fw[g])/(e-g)+(e-g)*(((((fy[u]-fx[e])/(u-e))-((fx[e]-fz[v])/(e
-v)))/(u-v))-((((fy[u]-fx[e])/(u-e))-((fx[e]-fw[g])/(e-g)))/(u
-g))-((((fz[v]-fx[e])/(v-e))-((fx[e]-fw[g])/(e-g)))/(v-g)));
e[n+1]=g-(fw[g]/j)//FullSimplify
```

Thus, the error of the root-solver in the $n+1-$iterate is attained in the following form and it shows that the iterative method (4) is of order fifteen

$$e_{n+1} = c_2^4 c_3^2 c_4(-3c_2^3 - 2c_2c_3 + c_4)e_n^{15} + O(e_n^{16}). \tag{5}$$

This ends the proof and shows that using the proposed approximation in the last step of the four-step cycle (1) increase the convergence order from eight to fifteen with adding only one more evaluation of the function per full iteration.

The idea used in (4) can be implemented at the end of any four-step cycle in which its first three steps are any of the optimal eighth-order methods to construct accurate and efficient four-step without memory iterations. That is to say by the approximation (3), we can build a class of four steps fifteenth-order methods. For example, using the method of Sharma and Sharma [12] in the first three steps and

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 3, No. 3 (2011)

the implementing a Newton's iteration with an approximation of the new appeared first derivative as in (3), we attain the following accurate fifteenth-order scheme

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - \dfrac{f(x_n)}{f(x_n) - 2f(y_n)} \dfrac{f(y_n)}{f'(x_n)}, \\ w_n = z_n - \dfrac{f(z_n) f[x_n, y_n]}{f[x_n, z_n] f[y_n, z_n]} (1 + \dfrac{f(z_n)}{f(x_n)}), \\ x_{n+1} = w_n - \dfrac{f(w_n)}{f[x_n, w_n] + (f[y_n, x_n, z_n] - f[y_n, x_n, w_n] - f[z_n, x_n, w_n])(x_n - w_n)}, \end{cases} \tag{6}$$

where its error equation is as comes next

$$e_{n+1} = c_2^4 (c_2^2 - c_3)^2 c_4 (3c_2^3 - 4c_2 c_3 + c_4) e_n^{15} + O(e_n^{16}). \tag{7}$$

Hence, by choosing any eighth-order method and using (3), we can build fifteenth-order without memory iterations using four function and one first derivative evaluations per full cycle. The (classical) efficiency index of our contributed schemes from the class is 1.7187 which is better than 1.4142 of Newton's method, 1.5874 of optimal two-step methods, 1.6817 of optimal three-step schemes and 1.6952 of Neta's fourteenth-order method. This index is just a little bit lower than that of optimal sixteenth-order methods 1.7411; but in contrast to such schemes, it possesses less computational burden.

3. Numerical Testing

This section deals with comparison of some numerical examples and obtaining the simple roots of the test problems. All the instances were done with Matlab 7.6. Unlike Section 2, we here use Matlab to show the readers that all of the iterative schemes can be implemented really well in all of the available Software. In examples considered in this article, the stopping criterion is the $|f(x_n)| \le \epsilon$, where $\epsilon = 10^{-2450}$. The absolute value of the given test functions after some full iterations are listed in Table 1. As Table 1 illustrates, the new methods from the class gives reliable results in all cases (when the computational time is taken into account as well), in contrast by the well-known methods with the same Total Number of Evaluations per cycle, i.e. Neta's fourteenth-order method (NM14), Neta and Petkovic's sixteenth-order method (NP16). The test functions are as follows:

$$\begin{array}{ll} f_1(x) = e^x + x - 20, & x^* \approx 2.842438953784447..., \\ f_2(x) = \sqrt{x^2 + 2x + 5} - 2\sin(x) - x^2 + 3, & x^* \approx 2.331967655883964..., \\ f_3(x) = 2x\cos(x) + x - 3, & x^* \approx -3.034664306974045..., \\ f_4(x) = (x-1)^6 - 1, & x^* = 2, \\ f_5(x) = \tan^{-1}(x), & x^* = 0. \end{array}$$

Fun., & x_0		NM14	NP16	(4)	(6)
$f_1, 3.5$	$\lvert f_1(x_1)\rvert$	0.2e-6	0.3e-7	0.4e-6	0.1e-7
	$\lvert f_1(x_2)\rvert$	0.4e-113	0.1e-143	0.1e-118	0.3e-141
	$\lvert f_1(x_3)\rvert$	0.1e-1608	0.8e-2327	0.1e-1806	0.3e-2147
	e-time	1.41	1.93	1.39	1.30
$f_2, 0.5$	$\lvert f_2(x_1)\rvert$	0.1e-7	0.5e-8	0.4e-8	0.6e-8
	$\lvert f_2(x_2)\rvert$	0.6e-124	0.2e-147	0.1e-141	0.1e-137
	$\lvert f_2(x_3)\rvert$	0.3e-1751	0.2e-2378	0.3e-2145	0.9e-2084
	e-time	2.20	2.71	2.13	2.08
$f_3, -3.2$	$\lvert f_3(x_1)\rvert$	0.5e-3	0.7e-3	0.7e-3	0.7e-4
	$\lvert f_3(x_2)\rvert$	0.3e-44	0.1e-47	0.3e-49	0.6e-63
	$\lvert f_3(x_3)\rvert$	0.7e-621	0.2e-762	0.9e-746	0.4e-948
	e-time	1.84	2.37	1.80	1.73
$f_4, 2.6$	$\lvert f_4(x_1)\rvert$	0.1	0.4e-1	0.2e-1	0.3e-1
	$\lvert f_4(x_2)\rvert$	0.3e-16	0.4e-27	0.1e-32	0.1e-28
	$\lvert f_4(x_3)\rvert$	0.3e-235	0.2e-443	0.1e-500	0.1e-439
	e-time	1.25	1.79	1.22	1.16
$f_5, 1$	$\lvert f_5(x_1)\rvert$	0.1e-3	0.8e-4	0.9e-5	0.5e-5
	$\lvert f_5(x_2)\rvert$	0.5e-69	0.5e-89	0.5e-108	0.4e-114
	$\lvert f_5(x_3)\rvert$	0.1e-1179	0.8e-1879	0.8e-2277	0.6e-2406
	e-time	1.33	1.86	1.28	1.21

Table 1: The comparison of some very high-order methods

From the results displayed in Table 1, it can be concluded that the proposed multi-point methods of the class are competitive with existing four-point four-step very high-order derivative-involved methods and possesses quick convergence for good initial approximations. The computer specifications are Intel(R) Core(TM) 2 Quad CPU, Q9550 @ 2.83GHz with 2.00GB of RAM. The e-time (elapsed time in seconds) for running the MATLAB Codes of each method for performing three full cycles has also been carried out using "tic-toc" command. The mean over 30 performances of the methods as the representatives to the iterations are also listed in Table 1.

We here remark that multi-point iterative schemes are so convenient for fast convergence when the initial guess is enough close to the exact root. If the guess point be far from the simple root then their convergence rates become slow. This is because, the multi-point iterative methods are predictor-corrector schemes in essence and the first step, i.e., the Newton's method does not guarantee the convergence. For circumventing on this drawback of all of the multi-point iterative methods, it is better off to use the non-iterative method of Yun [16] for finding a very appropriate initial estimation of the root.

4. Conclusion

In the language used so far, the new class of fifteenth-order schemes is a great improvement of the previously known methods and Table 1 shows that the iterative scheme is comparable with all of the methods. Per iteration this novel class requires four evaluations of the function and one evaluation of its first derivative, which implies that the efficiency index of the improved methods is 1.7187. More research by the approach of weight functions can be done in order to make the methods from our class optimal in the sense of Kung-Traub (1974).

References

[1] W. Bi, H. Ren, Q. Wu, Three-step iterative methods with eighth-order convergence for solving non-linear equations, *J. Comput. Appl. Math.,* **225** (2009), 105-112.

[2] M.T. Darvishi, Some three-step iterative methods free from second order derivative for finding solutions of systems of nonlinear equations, *Int. J. Pure Appl. Math.*, **57** (2009), 557-574.

[3] M.T. Darvishi, A two-step high order Newton-like method for solving systems of nonlinear equations, *Int. J. Pure Appl. Math.*, **57** (2009), 543-555.

[4] Y.H. Geum, Y.I. Kim, A multi-parameter family of three-step eighth-order iterative methods locating a simple root, *Appl. Math. Comput.,* **215** (2010), 3375-3382.

[5] Y.H. Geum, Y.I. Kim, A penta-parametric family of fifteenth-order multipoint methods for nonlinear equations, *Appl. Math. Comput.* **217** (2010), 2311-2319.

[6] S. Hilout, Superlinear convergence of a family of two-step Steffensen-type methods for generalized equations, *Int. J. Pure Appl. Math.,* **40** (2007), 1-10.

[7] R.F. King, Family of fourth-order methods for nonlinear equations, *SIAM J. Numer. Anal.,* **10** (1973), 876-879.

[8] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iteration, *J. Assoc. Comput. Mach.,* **21** (1974), 643-651.

[9] B. Neta, M.S. Petkovic, Construction of optimal order nonlinear solvers using inverse interpolation, *Appl. Math. Comput.* (2010). In Press.

[10] B. Neta, On a family of multipoint methods for non-linear equations, *Int. J. Comput. Math.* **9**, 353-361 (1981).

[11] T. Sauer, *Numerical Analysis,* Pearson, Boston, 2006.

[12] J.R. Sharma, R. Sharma, A new family of modified Ostrowski's methods with accelerated eighth order convergence, *Numer. Algorithms,* **54** (2010), 445-458.

[13] F. Soleymani, M. Sharifi, On a cubically iterative scheme for solving nonlinear equations, *Far East J. Appl. Math.,* **43** (2010), 137-143.

[14] F. Soleymani, Regarding the accuracy of optimal eighth-order methods, *Math. Comput. Modelling,* **53** (2011), 1351-1357.

[15] F. Soleymani, Concerning some sixth-order iterative methods for finding the simple roots of nonlinear equations, *Bull. Math. Anal. Appl.,* **2** (2010), 146-151.

[16] B.I. Yun, A non-iterative method for solving non-linear equations, *Appl. Math. Comput.,* **198** (2008), 691-699.