Research Article

Computing Simple Roots by an Optimal Sixteenth-Order Class

F. Soleymani,[1] S. Shateyi,[2] and H. Salmani[3]

[1] *Department of Mathematics, Islamic Azad University, Zahedan Branch, Zahedan, Iran*
[2] *Department of Mathematics, University of Venda, Private Bag X5050,*
  Thohoyandou 0950, South Africa
[3] *Department of Civil Engineering, Islamic Azad University, Zahedan Branch, Zahedan, Iran*

Correspondence should be addressed to S. Shateyi, stanford.shateyi@univen.ac.za

The problem considered in this paper is to approximate the simple zeros of the function $f(x)$ by iterative processes. An optimal 16th order class is constructed. The class is built by considering any of the optimal three-step derivative-involved methods in the first three steps of a four-step cycle in which the first derivative of the function at the fourth step is estimated by a combination of already known values. Per iteration, each method of the class reaches the efficiency index $\sqrt[5]{16} \approx 1.741$, by carrying out four evaluations of the function and one evaluation of the first derivative. The error equation for one technique of the class is furnished analytically. Some methods of the class are tested by challenging the existing high-order methods. The interval Newton's method is given as a tool for extracting enough accurate initial approximations to start such high-order methods. The obtained numerical results show that the derived methods are accurate and efficient.

1. Introduction

Consider that the function $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ is a sufficiently differentiable scalar function. We assume that $r \in I$ be a simple zero of the nonlinear function, that is, $f(r) = 0$ and $f'(r) \neq 0$. There is a vast literature on simple zeros of such nonlinear functions by iterative processes, see, for example, [1–3]. In 1974, Kung and Traub [4] conjectured that a multipoint iterative method without memory for solving single variable nonlinear equations consisting of $n + 1$ evaluations per iteration has the maximal convergence rate 2^n and, subsequently, the maximal efficiency index will be $\sqrt[n+1]{2^n}$. By taking into consideration this concept, many authors, see, for example, [5–7], have tried to produce optimal multistep methods.

Kung and Traub in [4] presented the following n-point optimal iterative process consisting of $(n + 1)$-evaluation per full cycle:

$$q_2(f)(x_n) = x_n - \frac{f(x_n)}{f'(x_n)},$$

$$\vdots$$

$$q_{j+1}(f)(x_n) = S_j(0), \tag{1.1}$$

for $j = 2, \ldots, n - 1$, where $S_j(y)$ is the inverse Hermite interpolatory polynomial of degree at most j such that

$$S_j(f(x_n)) = x_n, \qquad S'_j(f(x_n)) = \frac{1}{f'(x_n)},$$

$$S_j(f(q_\lambda(f)(x_n))) = q_\lambda(x_n), \quad \lambda = 2, \ldots, j. \tag{1.2}$$

This technique is defined by $x_{n+1} = q_n(f)(x_n)$ starting with an initial point x_0. Recently, Neta and Petković applied the concept of inverse interpolation again in [8] to approximate the first derivative of the function in the third and fourth steps of a three- and four-step cycle. They obtained the following optimal 8th order technique:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)},$$

$$z_n = y_n - \frac{f(x_n) + tf(y_n)}{f(x_n) + (t - 2)f(y_n)} \frac{f(y_n)}{f'(x_n)}, \tag{1.3}$$

$$x_{n+1} = y_n + c[f(x_n)]^2 - d[f(x_n)]^3,$$

with $t \in \mathbb{R}$, and $c, \ d$ as comes next

$$d = \frac{1}{(f(y_n) - f(x_n))(f(y_n) - f(z_n))f[y_n, x_n]} - \frac{1}{(f(z_n) - f(x_n))(f(y_n) - f(z_n))f[z_n, x_n]}$$

$$+ \frac{1}{f'(x_n)(f(z_n) - f(x_n))(f(y_n) - f(z_n))} - \frac{1}{f'(x_n)(f(y_n) - f(x_n))(f(y_n) - f(z_n))},$$

$$c = \frac{1}{(f(y_n) - f(x_n))f[y_n, x_n]} - \frac{1}{f'(x_n)(f(y_n) - f(x_n))} - d(f(y_n) - f(x_n)). \tag{1.4}$$

They also presented an optimal 16th order technique in the following structure consisting of four evaluations of the function and one evaluation of the first derivative:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)},$$

$$z_n = y_n - \frac{f(x_n) + tf(y_n)}{f(x_n) + (t-2)f(y_n)} \frac{f(y_n)}{f'(x_n)},$$

$$w_n = y_n + c[f(x_n)]^2 - d[f(x_n)]^3,$$

$$x_{n+1} = y_n + c'[f(x_n)]^2 - d'[f(x_n)]^3 + g'[f(x_n)]^4, \tag{1.5}$$

wherein (without the index n)

$$g' = \frac{(\varphi_w - \varphi_z)/(f(w) - f(z)) - ((\varphi_y - \varphi_z)/(f(y) - f(z)))}{f(w) - f(y)},$$

$$d' = \frac{\varphi_w - \varphi_z}{f(w) - f(z)} - g'(f(w) - 2f(x) + f(z)),$$

$$c' = \varphi_w - d'(f(w) - f(x)) - g'(f(w) - f(x))^2, \tag{1.6}$$

and also $\varphi_w = 1/(f[w,x](f(w) - f(x))) - 1/(f'(x_n)(f(w) - f(x)))$, $\varphi_z = 1/(f[z,x](f(z) - f(x))) - 1/(f'(x_n)(f(z) - f(x)))$, and $\varphi_y = 1/(f[y,x](f(y) - f(x))) - 1/(f'(x_n)(f(y) - f(x)))$.

For further reading, one may refer to [9, 10]. Contents of the paper are summarized in what follows. In the next section, our novel contribution is constructed by considering an optimal eighth-order method in the first three steps of a four-step cycle, in which the derivative in the quotient of the new Newton's iteration is estimated such that the order remains at 16, namely, optimal according to the Kung-Traub hypothesis. The new class of methods is supported with detailed proof in this section to verify the construction theoretically.

Then, it will be observed that the computed results listed in Table 2 completely support the theory of convergence and efficiency analyses discussed in the paper. Section 4 reminds the well-known *Interval Newton method* in the efficient programming package Mathematica 8 to extract enough accurate initial guesses to start the process. Finally, a short conclusion is given in the last section.

2. Main Results

In this section, we derive our main results by providing a class of four-step iterative methods, which agrees with Kung-Traub hypothesis. In order to build the class, we consider the

Table 1: Test functions and their zeros.

Nonlinear functions	Zeros
$f_1(x) = \sqrt{x^4 + 8}\sin(\pi/(x^2 + 2)) + x^3/(x^4 + 1) - \sqrt{6} + 8/17$	$r_1 = -2$
$f_2(x) = \sqrt{x^2 + 2x + 5} - 2\sin(x) - x^2 + 3$	$r_2 \approx 2.331967655883964$
$f_3(x) = \sin(x) - x/100$	$r_3 = 0$
$f_4(x) = (1/3)x^4 - x^2 - (1/3)x + 1$	$r_4 = 1$
$f_5(x) = e^{\sin(x)} - 1 - x/5$	$r_5 = 0$
$f_6(x) = xe^{x^2} - (\sin(x))^2 + 3\cos(x) + 5$	$r_6 \approx -1.207647827130919$
$f_7(x) = e^{-x} + \cos(x)$	$r_7 \approx 1.746139530408013$
$f_8(x) = x^4 + \sin(\pi/(x^2)) - 5$	$r_8 = \sqrt{2}$
$f_9(x) = 10xe^{-x^2} - 1$	$r_9 \approx 1.679630610428450$
$f_{10}(x) = x^3 + 4x^2 - 15$	$r_{10} \approx 1.631980805566063$

following four-step four-point iteration, in which the first three steps are any of the optimal three-step three-point without memory derivative-involved methods:

$$\text{optimal 8th-order method} = \begin{cases} f(x_n) \text{ and } f'(x_n) \text{ are available,} \\ f(y_n) \text{ is available,} \\ f(z_n) \text{ is available,} \end{cases} \qquad (2.1)$$

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

It can be seen that per full cycle of the structure (2.1), it includes four evaluations of the function and two evaluations of the first derivative to reach the convergence rate 16. At this time, we should approximate the first derivative of the function at the fourth step such that the convergence order does not decrease. In order to do this effectively, we have to use all of the known data from the past steps, that is, $f(x_n)$, $f'(x_n)$, $f(y_n)$, $f(z_n)$, and $f(w_n)$. Herein, we take into account the nonlinear fraction given by (without the index n)

$$p(t) = \frac{b_1 + b_2(t - x) + b_3(t - x)^2 + b_4(t - x)^3}{1 + b_5(t - x)}. \qquad (2.2)$$

This nonlinear fraction is inspired by Pade approximant in essence. The approximation function should meet the interpolation conditions $f(x_n) = p(x_n)$, $f'(x_n) = p'(x_n)$, $f(y_n) = p(y_n)$, $f(z_n) = p(z_n)$, and $f(w_n) = p(w_n)$. Note that the first derivative of (2.2) takes the following form:

$$p'(t) = \frac{b_2 - b_1b_5 + 2b_3(t - x_n) + (3b_4 + b_3b_5)(t - x_n)^2 + 2b_4b_5(t - x_n)^3}{(1 + b_5(t - x_n))^2}. \qquad (2.3)$$

Table 2: Results of comparisons for different methods.

Functions and x_0	G-K 16	N-P 16	S-S 16 (I)	S-S 16 (II)
f_1, $x_0 = -1.2$				
$\|f_1(x_1)\|$	$0.5e-12$	$0.9e-15$	$0.8e-17$	$0.1e-15$
$\|f_1(x_2)\|$	$0.3e-187$	$0.8e-232$	$0.3e-266$	$0.6e-248$
$\|f_1(x_3)\|$	$0.2e-2989$	$0.1e-3704$	0	$0.3e-3963$
f_1, $x_0 = -3$				
$\|f_1(x_1)\|$	$0.6e-6$	$0.6e-6$	$0.9e-8$	$0.2e-7$
$\|f_1(x_2)\|$	$0.4e-93$	$0.1e-92$	$0.2e-122$	$0.5e-115$
$\|f_1(x_3)\|$	$0.3e-1487$	$0.1e-1427$	$0.7e-1955$	$0.2e-1838$
f_2, $x_0 = 2$				
$\|f_2(x_1)\|$	$0.4e-18$	$0.3e-19$	$0.5e-17$	$0.7e-19$
$\|f_2(x_2)\|$	$0.6e-310$	$0.4e-327$	$0.6e-291$	$0.3e-321$
$\|f_2(x_3)\|$	0	0	0	0
f_2, $x_0 = 3$				
$\|f_2(x_1)\|$	$0.2e-11$	$0.7e-11$	$0.3e-12$	$0.4e-11$
$\|f_2(x_2)\|$	$0.1e-202$	$0.5e-193$	$0.3e-214$	$0.1e-196$
$\|f_2(x_3)\|$	$0.7e-3260$	$0.4e-3107$	$0.1e-3445$	$0.4e-3163$
f_3, $x_0 = 1.5$				
$\|f_3(x_1)\|$	0.4	0.3	0.5	0.3
$\|f_3(x_2)\|$	$0.1e-15$	$0.1e-10$	$0.1e-11$	$0.2e-17$
$\|f_3(x_3)\|$	$0.8e-262$	$0.2e-177$	$0.6e-197$	$0.4e-293$
f_3, $x_0 = -0.9$				
$\|f_3(x_1)\|$	F.	$0.1e-4$	$0.7e-7$	$0.3e-6$
$\|f_3(x_2)\|$	F.	$0.6e-109$	$0.2e-141$	$0.2e-142$
$\|f_3(x_3)\|$	F.	$0.1e-2298$	$0.1e-2695$	$0.2e-3001$
f_4, $x_0 = 0.5$				
$\|f_4(x_1)\|$	$0.4e-8$	$0.2e-8$	$0.1e-7$	$0.6e-9$
$\|f_4(x_2)\|$	$0.1e-129$	$0.4e-133$	$0.8e-122$	$0.8e-144$
$\|f_4(x_3)\|$	$0.4e-2073$	$0.1e-2129$	$0.5e-1950$	$0.1e-2302$
f_4, $x_0 = 1.5$				
$\|f_4(x_1)\|$	$0.5e-11$	$0.1e-12$	$0.2e-15$	$0.1e-13$
$\|f_4(x_2)\|$	$0.3e-175$	$0.1e-201$	$0.1e-245$	$0.6e-219$
$\|f_4(x_3)\|$	$0.8e-2803$	$0.4e-3225$	$0.2e-3930$	$0.9e-3503$
f_5, $x_0 = 1$				
$\|f_5(x_1)\|$	$0.1e-8$	$0.1e-6$	$0.2e-5$	$0.7e-8$
$\|f_5(x_2)\|$	$0.1e-139$	$0.1e-111$	$0.1e-86$	$0.1e-123$
$\|f_5(x_3)\|$	$0.9e-2238$	$0.6e-1787$	$0.4e-1304$	$0.2e-1860$
f_5, $x_0 = 4$				
$\|f_5(x_1)\|$	2.42	$0.6e-2$	$0.2e-1$	$0.3e-2$
$\|f_5(x_2)\|$	0.61	$0.1e-43$	$0.2e-33$	$0.1e-48$
$\|f_5(x_3)\|$	$0.2e-14$	$0.4e-712$	$0.5e-546$	$0.7e-790$
f_6, $x_0 = -2$				
$\|f_6(x_1)\|$	4.2	0.6	0.1	0.4
$\|f_6(x_2)\|$	$0.1e-8$	$0.3e-22$	$0.2e-30$	$0.3e-26$
$\|f_6(x_3)\|$	$0.5e-158$	$0.2e-377$	$0.9e-508$	$0.2e-443$
f_6, $x_0 = -0.6$				
$\|f_6(x_1)\|$	F.	$0.5e-1$	0.4	$0.1e-2$
$\|f_6(x_2)\|$	F.	$0.2e-38$	$0.4e-24$	$0.1e-66$

Table 2: Continued.

Functions and x_0	G-K 16	N-P 16	S-S 16 (I)	S-S 16 (II)
$\lvert f_6(x_3)\rvert$	F.	$0.6e-636$	$0.4e-408$	$0.1e-1088$
$f_7,\ x_0=0.5$				
$\lvert f_7(x_1)\rvert$	$0.3e-8$	$0.4e-8$	$0.4e-8$	$0.2e-8$
$\lvert f_7(x_2)\rvert$	$0.3e-142$	$0.9e-141$	$0.9e-142$	$0.2e-147$
$\lvert f_7(x_3)\rvert$	$0.7e-2288$	$0.2e-2264$	$0.3e-2280$	$0.4e-2372$
$f_7,\ x_0=3$				
$\lvert f_7(x_1)\rvert$	F.	1.4	$0.7e-1$	1.8
$\lvert f_7(x_2)\rvert$	F.	$0.4e-8$	$0.1e-27$	$0.1e-6$
$\lvert f_7(x_3)\rvert$	F.	$0.2e-141$	$0.2e-460$	$0.5e-117$
$f_8,\ x_0=1.1$				
$\lvert f_8(x_1)\rvert$	$0.4e-9$	$0.1e-9$	$0.4e-8$	$0.1e-9$
$\lvert f_8(x_2)\rvert$	$0.1e-162$	$0.4e-174$	$0.9e-151$	$0.2e-176$
$\lvert f_8(x_3)\rvert$	$0.1e-2621$	$0.5e-2806$	$0.1e-2432$	$0.3e-2843$
$f_8,\ x_0=2.5$				
$\lvert f_8(x_1)\rvert$	$0.4e-1$	$0.1e-1$	$0.4e-1$	$0.5e-2$
$\lvert f_8(x_2)\rvert$	$0.9e-36$	$0.6e-76$	$0.1e-37$	$0.1e-52$
$\lvert f_8(x_3)\rvert$	$0.1e-590$	$0.5e-755$	$0.1e-622$	$0.1e-863$
$f_9,\ x_0=0$				
$\lvert f_9(x_1)\rvert$	$0.1e-19$	$0.2e-18$	$0.3e-17$	$0.2e-19$
$\lvert f_9(x_2)\rvert$	$0.6e-335$	$0.1e-312$	$0.1e-294$	$0.1e-330$
$\lvert f_9(x_3)\rvert$	0	0	0	0
$f_9,\ x_0=2.2$				
$\lvert f_9(x_1)\rvert$	F.	0.1	2.4	$0.6e-1$
$\lvert f_9(x_2)\rvert$	F.	$0.3e-21$	$0.1e-6$	$0.4e-24$
$\lvert f_9(x_3)\rvert$	F.	$0.4e-349$	$0.2e-113$	$0.1e-395$
$f_{10},\ x_0=0.5$				
$\lvert f_{10}(x_1)\rvert$	F.	5.3	93	1.0
$\lvert f_{10}(x_2)\rvert$	F.	$0.1e-13$	$0.2e-2$	$0.9e-25$
$\lvert f_{10}(x_3)\rvert$	F.	$0.6e-246$	$0.2e-66$	$0.4e-425$
$f_{10},\ x_0=3$				
$\lvert f_{10}(x_1)\rvert$	$0.2e-3$	$0.1e-3$	$0.4e-4$	$0.3e-4$
$\lvert f_{10}(x_2)\rvert$	$0.4e-80$	$0.1e-86$	$0.1e-95$	$0.3e-96$
$\lvert f_{10}(x_3)\rvert$	$0.5e-1309$	$0.2e-1412$	$0.1e-1560$	$0.5e-1568$

Substituting the known data in (2.2) and (2.3) will result in obtaining the five unknown parameters. It is obvious that $b_1 = f(x_n)$, hence we obtain the following system of four linear equations with four unknowns:

$$b_2 - f(x_n)b_5 = f'(x_n),$$

$$(y_n - x_n)b_2 + (y_n - x_n)^2 b_3 + (y_n - x_n)^3 b_4 - (y_n - x_n)f(y_n)b_5 = f(y_n) - f(x_n),$$

$$(z_n - x_n)b_2 + (z_n - x_n)^2 b_3 + (z_n - x_n)^3 b_4 - (z_n - x_n)f(z_n)b_5 = f(z_n) - f(x_n),$$

$$(w_n - x_n)b_2 + (w_n - x_n)^2 b_3 + (w_n - x_n)^3 b_4 - (w_n - x_n)f(w_n)b_5 = f(w_n) - f(x_n).$$

$$(2.4)$$

Solving (2.4) and simplifying (without the index n and using divided differences) yields in

b_5

$$= \frac{(f[z,x]WY(w-y) + Z(f[w,x](y-z)Y + f[y,x](z-w)W) - (w-y)(w-z)(y-z)f'(x))}{(YZ(z-y)f(w) + (w-z)((w-y)(y-z)f(x) + WZf(y)) + WY(y-w)f(z))},$$

$$b_2 = f'(x) + f(x)b_5,$$

$$b_4 = \frac{(z-x)f[y,x] + (x-y)f[z,x] + (y-z)b_2 + ((z-x)f(y) + (x-y)f(z))b_5}{(x-y)(x-z)(y-z)},$$

$$b_3 = f[w,x,x] + f[w,x]b_5 - (w-x)b_4,$$

$$(2.5)$$

wherein $y - x = Y$, $z - x = Z$, $w - x = W$. Now we have a powerful approximation of the first derivative of the function in the fourth step of (2.1), which doubles the convergence rate of the optimal 8th order methods. Therefore, we attain the following class, in which we have four evaluations of the function and one evaluation of the first-order derivative:

$$\text{Any optimal 8th-order method} = \begin{cases} f(x_n), \text{ and } f'(x_n) \text{ are available,} \\ f(y_n) \text{ is available,} \\ f(z_n) \text{ is available,} \end{cases}$$

$$x_{n+1} = w_n - \frac{(1 + b_5(w_n - x_n))^2 f(w_n)}{f'(x_n) + 2b_3(w_n - x_n) + (3b_4 + b_3b_5)(w_n - x_n)^2 + 2b_4b_5(w_n - x_n)^3}.$$

$$(2.6)$$

Now using any of the optimal three-point three-step methods, we could obtain a novel 16th order technique which satisfies the Kung-Traub hypothesis as well. Using the optimal 8th order method given by Wang and Liu in [11] ends in the following four-step method:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)},$$

$$z_n = x_n - \frac{f(x_n)}{f'(x_n)} \frac{f(x_n) - f(y_n)}{f(x_n) - 2f(y_n)},$$

$$w_n = z_n - \frac{f(z_n)}{f'(x_n)}\left[\frac{1}{2} + \frac{5f(x_n)^2 + 8f(x_n)f(y_n) + 2f(y_n)^2}{5f(x_n)^2 - 12f(x_n)f(y_n)}\left(\frac{1}{2} + \frac{f(z_n)}{f(y_n)}\right)\right],$$

$$x_{n+1} = w_n - \frac{(1 + b_5(w_n - x_n))^2 f(w_n)}{f'(x_n) + 2b_3(w_n - x_n) + (3b_4 + b_3b_5)(w_n - x_n)^2 + 2b_4b_5(w_n - x_n)^3}.$$

$$(2.7)$$

Theorem 2.1. *Assume that the scalar function f be sufficiently smooth in the real open domain $I \subseteq \mathbb{R}$. Furthermore let $e_n = x_n - r$ and $c_k = (f^{(k)}(r))/(k!f'(r))$, for $k \in N$, N is the set of natural numbers.*

Then, the iterative method (2.7) *has the optimal order of convergence 16 and satisfies the following error equation:*

$$e_{n+1} = \frac{1}{25c_3}\left[c_2^3\left(c_2^2 - c_3\right)^2 \left(11c_2^4 - 25c_2^2c_3 + 5c_3^2 + 5c_2c_4\right) \right.$$

$$\left. \times \left(11c_2^4c_3 - 25c_2^2c_3^2 + 5c_2c_3c_4 + 5\left(c_3^3 + c_4^2 - c_3c_5\right)\right)\right]e_n^{16} + O\left(e_n^{17}\right). \tag{2.8}$$

Proof. Using Taylor series expansion around the simple root r in the nth iterate, [12], results in $f(x_n) = f'(r)[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + \cdots + O(e_n^{17})]$, and also $f'(x_n) = f'(r)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + \cdots + O(e_n^{16})]$. Applying these expansions in the first step of (2.7), we have

$$y_n - r = c_2e_n^2 + \left(-2c_2^2 + 2c_3\right)e_n^3 + \left(4c_2^3 - 7c_2c_3 + 3c_4\right)e_n^4 + \cdots + O\left(e_n^{17}\right). \tag{2.9}$$

Note that to keep the prerequisites at the minimum, we only mention the simplified error equations at the end of first, second, and third steps. Because we also know that the first three steps are an eighth-order technique. Taylor series expanding in the second step of (2.7) by applying (2.9) yields in

$$z_n - r = \left(c_2^3 - c_2c_3\right)e_n^4 - 2\left(2c_2^4 - 4c_2^2c_3 + c_3^2 + c_2c_4\right)e_n^5 + \cdots + O\left(e_n^{17}\right). \tag{2.10}$$

Using (2.10) and the third step of (2.7) gives us

$$w_n - r = \frac{1}{5}c_2\left(c_2^2 - c_3\right)\left(11c_2^4 - 25c_2^2c_3 + 5c_3^2 + 5c_2c_4\right)e_n^8 + \cdots + O\left(e_n^{17}\right). \tag{2.11}$$

Now, we expand $f(w_n)$, we obtain

$$f(w_n) = \frac{1}{5}c_2\left(c_2^2 - c_3\right)\left(11c_2^4 - 25c_2^2c_3 + 5c_3^2 + 5c_2c_4\right)f'(r)e_n^8 + \cdots + O\left(e_n^{17}\right). \tag{2.12}$$

Subsequently, in the last step we have

$$\frac{(1 + b_5(w_n - x_n))^2 f(w_n)}{f'(x_n) + 2b_3(w_n - x_n) + (3b_4 + b_3b_5)(w_n - x_n)^2 + 2b_4b_5(w_n - x_n)^3}$$

$$= \frac{1}{5}c_2\left(c_2^2 - c_3\right)\left(11c_2^4 - 25c_2^2c_3 + 5c_3^2 + 5c_2c_4\right)e_n^8 + \cdots + O\left(e_n^{17}\right). \tag{2.13}$$

Finally, using (2.12) and (2.13) in the last step of (2.7) ends in

$$
e_{n+1} = x_{n+1} - r = \frac{1}{25c_3}\left[c_2^3\left(c_2^2 - c_3\right)^2\left(11c_2^4 - 25c_2^2c_3 + 5c_3^2 + 5c_2c_4\right)\right.
$$
$$
\left. \times\left(11c_2^4c_3 - 25c_2^2c_3^2 + 5c_2c_3c_4 + 5\left(c_3^3 + c_4^2 - c_3c_5\right)\right)\right]e_n^{16} + O\left(e_n^{17}\right),
$$
$$(2.14)$$

which shows that (2.7) is a 16th order method consuming only five evaluations per iteration. This completes the proof. $\square$

Remark 2.2. If we choose any of the other optimal 8th order derivative-involved methods in the first three steps of (2.6), then a novel optimal 16th order technique will be obtained. For instance, using the optimal 8th order method given by J. R. Sharma and R. Sharma in [13] results in the following four-step optimal method:

$$
y_n = x_n - \frac{f(x_n)}{f'(x_n)},
$$
$$
z_n = y_n - \frac{f(x_n)}{f(x_n) - 2f(y_n)}\frac{f(y_n)}{f'(x_n)},
$$
$$
w_n = z_n - \left[1 + \frac{f(z_n)}{f(x_n)}\right]\frac{f[x_n, y_n]f(z_n)}{f[y_n, z_n]f[x_n, z_n]},
$$
$$
x_{n+1} = w_n - \frac{(1 + b_5(w_n - x_n))^2 f(w_n)}{f'(x_n) + 2b_3(w_n - x_n) + (3b_4 + b_3b_5)(w_n - x_n)^2 + 2b_4b_5(w_n - x_n)^3},
$$
$$(2.15)$$

which satisfies the following error equation:

$$
e_{n+1} = \frac{c_2^4}{c_3}\left[\left(c_2^2 - c_3\right)^2\left(3c_2^3 - 4c_2c_4 + c_4\right)\left(3c_2^4c_3 - 4c_2^2c_3^2 + c_2c_3c_4 + c_4^2 - c_3c_5\right)\right]e_n^{16} + O\left(e_n^{17}\right).
$$
$$(2.16)$$

Remark 2.3. Any method of the developed class carries out five evaluations per full cycle to reach the optimal order 16. Hence, the index of efficiency for our class is $\sqrt[5]{16} \approx 1.741$, which is greater than that of optimal 4th order techniques $\sqrt[3]{4} \approx 1.587$, and optimal 8th order techniques $\sqrt[4]{8} \approx 1.682$.

3. Computational Experiments

The analytical outcomes given in the last section are fully supported by numerical experiments here. Two methods of the class (2.6), that is, methods (2.7) (S-S 16 (I)) and (2.15) (S-S 16 (II)), are compared with some of high-order methods.

For numerical comparisons, we have the 16th-order method of Geum and Kim [14] given as follows:

$$y_n = x_n - f'(x_n)^{-1} f(x_n),$$

$$z_n = y_n - K_f f'(x_n)^{-1} f(y_n),$$

$$w_n = z_n - H_f f'(x_n)^{-1} f(z_n),$$

$$x_{n+1} = w_n - W_f f'(x_n)^{-1} f(w_n),$$

(3.1)

wherein $K_f = (1 + 2u - 4u^2)/(1 - 3u^2)$, $H_f = (1 + 2u)/(1 - v - 2q)$, $W_f = u(1 - 6u - u^2 - 24u^3)q + 2(1 - u)q^2 + (1 + 2u)/(1 - t - v - 2q - 2vq)$, and $u = f(y)/f(x)$, $v = f(z)/f(y)$, $q = f(z)/f(x)$, $t = f(w)/f(z)$ (without the index n). This method is denoted by (G-K 16). We also used the 16th order method obtained by Neta and Petković (1.5) with $t = 0$ (N-P 16). We do not include other schemes with other orders, and one may refer to [15–21] for having further information about the other recent methods. Among many numerical examples, we have selected ten instances, with the use of multiprecision arithmetic. The test functions are displayed in Table 1.

For comparisons, 4000 digits floating points have been considered. As it shows, any method of the presented class is more efficient than the existing high order methods in the literature. Results in Table 2 manifest the applicability of the new scheme in the test problems. Also notice that F. stands for failure, that is, the starting value is not sufficiently close to the zero to make the iterative method converge.

We have checked that the sequence of iterations converge to an approximation of the solution for nonlinear equations in our numerical works. We note that an important problem of determining good starting points appears, when applying iterative methods for solving nonlinear equations. A quick convergence, one of the advantages of multipoint methods, can be attained only if initial approximations are sufficiently close to the sought roots; otherwise, it is not possible to realize the expected convergence speed in practice. For achieving such a goal, we remind the well-known technique of interval Newton's method as a tool to obtain robust initial approximations in the next section.

4. All the Zeros

In this section, we pay close attention to the matter of finding all the (simple) zeros of nonlinear functions in an interval using a hybrid algorithm. In fact, in practice one wish to find all the zeros at the same time. On the other hand, the iterative scheme, such as (2.7), is so sensitive upon the choice of the initial guess. Generally speaking, no method is the best iterative scheme in all test problems. Mostly, this is due to the test function and the accuracy of the initial approximation.

Having a sharp enough initial estimation, one may start the process as efficiently as possible and to see the computational order of convergence correctly, that is, the rate of correcting the number decimal places to the true solution. However, having such an initial guess is not always an easy task for practical problems.

In what follows, we remind a well-known scheme based on [22], also known as interval Newton's method for finding enough accurate initial approximations for all the zeros

of a nonlinear function in an interval. Hopefully, Mathematica 8 gives the users a friendly environment for working with lists (sequences), that is to say, the obtained list then would be corrected using the new 16th order schemes of this paper.

This procedure was efficiently coded in [22] and with some changes we have provided in what follows:

```
intervalnewton::rec = "MaxRecursion exceeded.";
intnewt[f_,df_,x_,{a_,b_},eps_,n_]:=
Block[{xmid,int=Interval[{a,b}]},If[b-a<eps,Return[int]];
    If[n==0,Message[intervalnewton::rec];
     Return[int]];
    xmid=Interval[SetAccuracy[(a+b)/2,16]];
    int=IntervalIntersection[int,SetAccuracy[xmid-
N[f/.x->xmid]/N[df/.x->int],16]];
  (intnewt[f,df,x,#,eps,n-1])&/@(List@@int)];
Options[intervalnewton]={MaxRecursion->2000};
```

The above piece of code takes the nonlinear function f, the lower and upper bounds of the working interval and applies the interval form of the Newton's iteration with the maximum number of recursion considered as 2000 and also the use of the command `SetAccuracy[exp,16]`. Note that in this case, like the Newton's method, the function should have first order derivative on the interval. Thus now, the call-out function to implement the above piece of code could be given by

```
intervalnewton[f_,x_,int_Interval,eps_,opts___]
:=Block[{df,n},n=MaxRecursion/.{opts}/.Options[intervalnewton];
    df=D[f,x];
IntervalUnion@@Select[Flatten[(intnewt[f,df,x,#,eps,n])&/@(List@@int)],
IntervalMemberQ [f/.x->#,0]&]];
```

The tolerance should be chosen to be arbitrary in machine precision. To illustrate the procedure further, we consider the oscillatory nonlinear function $f(x) = e^{\sin(\log(x)\cos(20x))} - 2$, where its plot has been given in Figure 1, in the interval $[a, b] = [2., 10.]$ while the tolerance is set to 0.0001 as follows:

```
f[x_]:=Exp[Sin[Log[x]*Cos[20x]]]-2;
IntervalSol=intervalnewton[f[x],x,Interval[{2.,10.}],.0001];
setInitial=N[Mean/@List@@IntervalSol]
NumberOfGuesses=Length[setInitial];
```

The implementation of the above part of code shows that the number of zeros is 51 while the list of initial approximations is {2.18856, 2.2125, 2.48473, 2.54372, 2.791, 2.86524, 3.10023, 3.18401, 3.41091, 3.50142, 3.72247, 3.81803, 4.03459, 4.13411, 4.3471, 4.44984, 4.65989, 4.7653, 4.9729, 5.08056, 5.28606, 5.39566, 5.59937, 5.71065, 5.91277, 6.02554, 6.22625, 6.34035, 6.53981, 6.65509, 6.85342, 6.96977, 7.16709, 7.28441, 7.4808, 7.599, 7.79455, 7.91356, 8.10833,
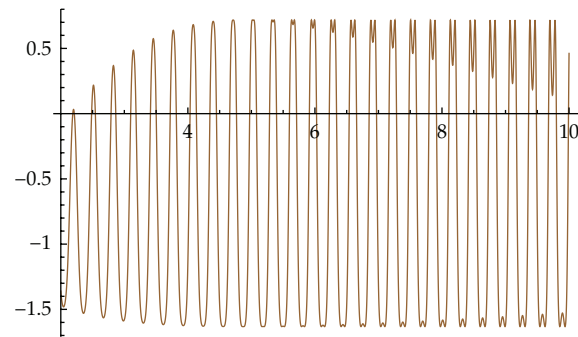
Figure 1: The plot of the function f with finitely many zeros in the interval $[2, 10]$.

8.22809, 8.42214, 8.54259, 8.73598, 8.85707, 9.04983, 9.17152, 9.36371, 9.48595, 9.67761, 9.80037, and 9.99152}.

We should remark that now by considering the new optimal 16th order methods from the class (2.6), one may enrich the accuracy of the initial guesses up to the desired tolerance when high precision computing is needed.

5. Conclusions

A class of four-point four-step iterative methods has been developed for solving nonlinear equations. The analytical proof for one method of the class was written to clarify the 16th order convergence. The class was attained by approximating the first derivative of the function in the fourth step of a cycle in which the first three steps are any of the optimal eighth-order derivative-involved methods.

Per full cycle, the methods of the class consist of four evaluations of the function and one evaluation of the first derivative which results in 1.741 as the efficiency index. The presented class satisfies the still unproved conjecture of Kung-Traub on multi-point iterations without memory.

The accuracy and efficiency of two obtained methods of the class were illustrated by solving a lot of numerical examples. Numerical works also have attested the theoretical results given in the paper and have shown the fast rate of convergence.

Acknowledgment

References

[1] T. Sauer, *Numerical Analysis*, Pearson, Boston, Mass, USA, 2nd edition, 2011.

[2] F. Soleymani, "Optimized Steffensen-type methods with eighth-order convergence and high efficiency index," *International Journal of Mathematics and Mathematical Sciences*, vol. 2012, Article ID 932420, 18 pages, 2012.

[3] M. Heydari, S. M. Hosseini, and G. B. Loghmani, "Convergence of a family of third-order methods free from second derivatives for finding multiple roots of nonlinear equations," *World Applied Sciences Journal*, vol. 11, pp. 507–512, 2010.

[4] H. T. Kung and J. F. Traub, "Optimal order of one-point and multipoint iteration," *Journal of the Association for Computing Machinery*, vol. 21, pp. 643–651, 1974.

[5] F. Soleymani, "Optimal fourth-order iterative methods free from derivatives," *Miskolc Mathematical Notes*, vol. 12, no. 2, pp. 255–264, 2011.

[6] F. Soleymani, R. Sharma, X. Li, and E. Tohidi, "An optimized derivative-free form of the Potra-Ptak method," *Mathematical and Computer Modelling*, vol. 56, pp. 97–104, 2012.

[7] F. Soleymani and B. S. Mousavi, "On novel classes of iterative methods for solving nonlinear equations," *Computational Mathematics and Mathematical Physics*, vol. 52, no. 2, pp. 214–221, 2012.

[8] B. Neta and M. S. Petković, "Construction of optimal order nonlinear solvers using inverse interpolation," *Applied Mathematics and Computation*, vol. 217, no. 6, pp. 2448–2455, 2010.

[9] Y. H. Geum and Y. I. Kim, "A family of optimal sixteenth-order multipoint methods with a linear fraction plus a trivariate polynomial as the fourth-step weighting function," *Computers & Mathematics with Applications*, vol. 61, no. 11, pp. 3278–3287, 2011.

[10] F. Soleymani, M. Sharifi, and B. S. Mousavi, "An improvement of Ostrowski's and King's techniques with optimal convergence order eight," *Journal of Optimization Theory and Applications*, vol. 153, no. 1, pp. 225–236, 2012.

[11] X. Wang and L. Liu, "New eighth-order iterative methods for solving nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 234, no. 5, pp. 1611–1620, 2010.

[12] S. Wolfram, *The Mathematica Book*, Wolfram Media, 5th edition, 2003.

[13] J. R. Sharma and R. Sharma, "A new family of modified Ostrowski's methods with accelerated eighth order convergence," *Numerical Algorithms*, vol. 54, no. 4, pp. 445–458, 2010.

[14] Y. H. Geum and Y. I. Kim, "A biparametric family of optimally convergent sixteenth-order multipoint methods with their fourth-step weighting function as a sum of a rational and a generic two-variable function," *Journal of Computational and Applied Mathematics*, vol. 235, no. 10, pp. 3178–3188, 2011.

[15] Y. H. Geum and Y. I. Kim, "A biparametric family of four-step sixteenth-order root-finding methods with the optimal efficiency index," *Applied Mathematics Letters*, vol. 24, no. 8, pp. 1336–1342, 2011.

[16] H. Montazeri, F. Soleymani, S. Shateyi, and S. S. Motsa, "On a new method for computing the numerical solution of systems of nonlinear equations," *Journal of Applied Mathematics*, vol. 2012, Article ID 751975, 15 pages, 2012.

[17] Y. I. Kim and C. Chun, "New twelfth-order modifications of Jarratt's method for solving nonlinear equations," *Studies in Nonlinear Sciences*, vol. 1, pp. 14–18, 2010.

[18] Y. I. Kim, C. Chun, and W. Kim, "Some third-order curvature based methods for solving nonlinear equations," *Studies in Nonlinear Sciences*, vol. 1, pp. 72–76, 2010.

[19] F. Soleymani and S. K. Khattri, "Finding simple roots by seventh- and eighth-order derivative-free methods," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 6, pp. 45–52, 2012.

[20] F. Soleymani, "Optimal eighth-order simple root-finders free from derivative," *WSEAS Transactions on Information Science and Applications*, vol. 8, pp. 293–299, 2011.

[21] F. Soleymani and F. Soleimani, "Novel computational derivative-free methods for simple roots," *Fixed Point Theory*, vol. 13, no. 1, pp. 247–258, 2012.

[22] J. B. Keiper, "Interval arithmetic in mathematica," *The Mathematica Journal*, vol. 5, pp. 66–71, 1995.