

**NEW ITERATIVE METHODS
FOR SINGLE VARIABLE EQUATIONS**

Chahn Yong Jung¹, Arif Rafiq², Faisal Ali³, Shin Min Kang⁴ §

¹Department of Business Administration
Gyeongsang National University
Jinju, 52828, KOREA

²Department of Mathematics and Statistics
Virtual University of Pakistan
Lahore, 54000, PAKISTAN

³Center for Advanced Studies in Pure and Applied Mathematics
Bahauddin Zakariya University
Multan, 60800, PAKISTAN

⁴Department of Mathematics and RINS
Gyeongsang National University
Jinju, 52828, KOREA

Abstract: In this paper, we establish new iterative methods for the solution of scalar equations by using the decomposition technique due to Noor and Noor [Some iterative schemes for nonlinear equations, *Appl. Math. Comput.* **183** (2006), 774-779].

AMS Subject Classification: 65H05

Key Words: iterative methods, nonlinear equations, convergence order, multiple roots

1. Introduction

One of the most significant problems in numerical analysis is to find the solution of nonlinear equations. Various iterative methods have been studied to find the approximate roots of nonlinear equations, see [1, 3, 4, 5, 7, 10, 11, 13] and references there in. These methods can be classified as one-step, two-step and

Received: December 8, 2015

Published: March 4, 2016

© 2016 Academic Publications, Ltd.

url: www.acadpubl.eu

§Correspondence author

three-step methods. In [5], Chun has proposed several one-step and two-step iterative methods with higher-order convergence by using the decomposition technique of Adomian [2]. Many other iterative methods have also been developed for finding the simple roots of nonlinear equations. But there are many scientific problems in which the nature of the solution to the governing nonlinear equations is not simple. In order to find the multiple roots, Schröder [18] made the first contribution and introduced the following modified Newton's method:

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

If $m = 1$, then the equation is the classical Newton's formula.

During the last two decades, much attention has been devoted by various researchers for solving nonlinear equation with multiple roots. Chun et al. [6], Chun and Neta [8], Homeier [12], Osada [17] have developed some techniques to find the multiple roots of nonlinear equations. In the recent years, the researchers have made significant and interesting contribution in this field [9, 14, 15].

In this paper, we propose some new iterative methods for finding multiple roots of scalar equations by using decomposition technique given in [16].

Consider the nonlinear equation

$$f(x) = 0. \quad (1.1)$$

It is well known that if α is a root with multiplicity m , then it is also a root of $f(x) = 0$ with multiplicity $m - 1$ of $f'(x) = 0$ with multiplicity $m - 2$ and so on. Hence if initial guess x_0 is sufficiently close to α , the expressions

$$\begin{aligned} x_0 - m \frac{f(x_0)}{f'(x_0)}, \\ x_0 - (m - 1) \frac{f(x_0)}{f''(x_0)}, \\ x_0 - (m - 2) \frac{f(x_0)}{f'''(x_0)}, \\ \dots \end{aligned} \quad (1.2)$$

will have the same value.

Remark 1.1. The generalized Newton's formula

$$x_{n+1} = x_n - 2 \frac{f(x_n)}{f'(x_n)}, \quad (1.3)$$

gives a quadratic convergence when the equation $f(x) = 0$ has a pair of double roots in the neighbourhood of x_0 . It may be noted that for the double root α near to x_0 , $f(\alpha) = 0 = f'(\alpha)$.

2. New Iterative Methods

We can rewrite the nonlinear equation (1.1) as a coupled system:

$$f(\gamma) + (x - \gamma) \frac{f(\gamma) + f(x)}{2} + g(x) = 0, \quad (2.1)$$

$$g(x) = f(x) - f(\gamma) - (x - \gamma) \frac{f(\gamma) + f(x)}{2}, \quad (2.2)$$

where γ is the initial approximation for a zero of (1.1).

We can rewrite (2.1) in the following form:

$$\begin{aligned} x &= \gamma - 2 \frac{f(\gamma)}{f(\gamma)} - (x - \gamma) \frac{f(x)}{f(\gamma)} - 2 \frac{g(x)}{f(\gamma)} \\ &= c + N(x), \end{aligned} \quad (2.3)$$

where

$$c = \gamma - 2 \frac{f(\gamma)}{f(\gamma)}, \quad (2.4)$$

and

$$N(x) = -(x - \gamma) \frac{f(x)}{f(\gamma)} - 2 \frac{g(x)}{f(\gamma)}. \quad (2.5)$$

Here $N(x)$ is a nonlinear operator.

As in [16], the solution of (2.3) has the series form,

$$x = \sum_{i=0}^{\infty} x_i. \quad (2.6)$$

The nonlinear operator $N(x)$ can be decomposed as it has been shown in [16]. Also the series $\sum_{i=0}^{\infty} x_i$ converges absolutely and uniformly to a unique solution of equation (2.3) if the nonlinear operator

$$N(x) = N\left(\sum_{i=0}^{\infty} x_i\right) = N(x_0) + \sum_{i=1}^{\infty} \left\{ N\left(\sum_{j=0}^i x_j\right) \right\} \quad (2.7)$$

is a contraction.

Combining (2.3), (2.6) and (2.7), we have

$$\sum_{i=0}^n x_i = c + N(x_0) + \sum_{i=1}^n \left\{ N \left(\sum_{j=0}^i x_j \right) \right\}. \quad (2.8)$$

Thus we have the following iterative scheme:

$$\begin{aligned} x_0 &= c, \\ x_1 &= N(x_0), \\ x_2 &= N(x_0 + x_1), \\ &\vdots \\ x_{n+1} &= N(x_0 + x_1 + \dots + x_n), \quad n = 1, 2, \dots \end{aligned} \quad (2.9)$$

Then

$$\begin{aligned} x_1 + x_2 + \dots + x_{n+1} \\ = N(x_0) + N(x_0 + x_1) + \dots + N(x_0 + x_1 + \dots + x_n), \quad n = 1, 2, \dots, \end{aligned}$$

and

$$x = c + \sum_{i=1}^n x_i. \quad (2.10)$$

From (2.4), (2.5) and (2.9), we have

$$x_0 = c = \gamma - 2 \frac{f(\gamma)}{f'(\gamma)}, \quad (2.11)$$

and

$$\begin{aligned} x_1 &= N(x_0) \\ &= -(x_0 - \gamma) \frac{f'(x_0)}{f'(\gamma)} - 2 \frac{g(x_0)}{f'(\gamma)} \\ &= -2 \frac{f(x_0)}{f'(\gamma)}. \end{aligned} \quad (2.12)$$

It follows from (2.4), (2.9) and (2.10), that

$$\begin{aligned} x &\approx x_0 \\ &= c \\ &= \gamma - 2 \frac{f(\gamma)}{f'(\gamma)}. \end{aligned} \quad (2.13)$$

This enables us to suggest the following method for solving the nonlinear equation (1.1).

Algorithm 2.1. For the given x_0 compute the approximate solution x_{n+1} by the iterative schemes:

$$x_{n+1} = x_n - 2 \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0, \quad n = 0, 1, 2, \dots,$$

which is known as the generalized Newton's formula and is quadratically convergent.

Again by using (2.4), (2.5), (2.9) and (2.10), we conclude that

$$\begin{aligned} x &\approx x_0 + x_1 \\ &= c + x_1 \\ &= x_0 + N(x_0) \\ &= \gamma - 2 \frac{f(\gamma)}{f'(\gamma)} - 2 \frac{f(x_0)}{f'(\gamma)}. \end{aligned} \tag{2.14}$$

Using (2.14), we can suggest the following two-step iterative method for solving nonlinear equation (1.1) as:

Algorithm 2.2. For the given x_0 compute the approximate solution x_{n+1} by the iterative schemes:

$$\begin{aligned} y_n &= x_n - 2 \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0, \quad n = 0, 1, 2, \dots, \\ x_{n+1} &= y_n - 2 \frac{f(y_n)}{f'(x_n)}. \end{aligned}$$

Again, using (2.5), (2.11) and (2.12), we can calculate

$$\begin{aligned} N(x_0 + x_1) &= -(x_0 + x_1 - \gamma) \frac{f'(x_0 + x_1)}{f'(\gamma)} - 2 \frac{g(x_0 + x_1)}{f'(\gamma)} \\ &= -2 \frac{f(x_0)}{f'(\gamma)} - 2 \frac{f(x_0 + x_1)}{f'(\gamma)}. \end{aligned} \tag{2.15}$$

From (2.10), (2.11), (2.12) and (2.15), we get

$$\begin{aligned} x &\approx x_0 + x_1 + x_2 \\ &= c + N(x_0) + N(x_0 + x_1) \\ &= \gamma - 2 \frac{f(\gamma)}{f'(\gamma)} - 4 \frac{f(x_0)}{f'(\gamma)} - 2 \frac{f(x_0 + x_1)}{f'(\gamma)}. \end{aligned} \tag{2.16}$$

Using (2.16), we can suggest and analyze the following three-step iterative method for solving nonlinear equation (1.1).

Algorithm 2.3. For a given x_0 , compute the approximate solution by the iterative schemes:

$$\begin{aligned}
 y_n &= x_n - 2 \frac{f(x_n)}{f'(x_n)}, \quad f(x_n) \neq 0, \quad n = 0, 1, 2, \dots, \\
 z_n &= -2 \frac{f(y_n)}{f'(x_n)}, \\
 x_{n+1} &= y_n - 4 \frac{f(y_n)}{f'(x_n)} - 2 \frac{f(y_n + z_n)}{f'(x_n)}.
 \end{aligned}$$

3. Convergence Analysis

The convergence analysis of Algorithms 2.2 and 2.3 is given in this section.

Theorem 3.1. Assume that the function $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$ for an open interval D has a multiple root $\alpha \in D$ of multiplicity 2. Let $f(x)$ be sufficiently smooth in the neighborhood of the root α . Then the order of convergence of the methods defined by Algorithms 2.2 and 2.3 is 2.

Proof. Let α be a root of $f(x)$ of multiplicity 2, then by expanding $f(x_n)$ and $f'(x_n)$, in Taylor’s series about α , we obtain

$$f(x_n) = c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + O(e_n^5), \tag{3.1}$$

$$f'(x_n) = 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4 + O(e_n^5), \tag{3.2}$$

where $e_n = x_n - \alpha$ and $c_k = \frac{f^{(k)}(\alpha)}{k!}$, $k = 2, 3, \dots$

Using (3.1) and (3.2), we have

$$\begin{aligned}
 \frac{f(x_n)}{f'(x_n)} &= \frac{1}{2} e_n - \frac{1}{4} \frac{c_3}{c_2} e_n^2 + \left(-\frac{1}{2} \frac{c_4}{c_2} + \frac{3}{8} \frac{c_3^2}{c_2^2} \right) e_n^3 \\
 &+ \left(-\frac{3}{4} \frac{c_5}{c_2} + \frac{5}{4} \frac{c_3 c_4}{c_2^2} - \frac{9}{16} \frac{c_3^3}{c_2^3} \right) e_n^4 + O(e_n^5).
 \end{aligned} \tag{3.3}$$

Thus

$$\begin{aligned}
 y_n &= x_n - 2 \frac{f(x_n)}{f'(x_n)} \\
 &= \alpha + \frac{1}{2} \frac{c_3}{c_2} e_n^2 + \left(\frac{c_4}{c_2} - \frac{3}{4} \frac{c_3^2}{c_2^2} \right) e_n^3 \\
 &+ \left(\frac{3}{2} \frac{c_5}{c_2} - \frac{5}{2} \frac{c_3 c_4}{c_2^2} + \frac{9}{8} \frac{c_3^3}{c_2^3} \right) e_n^4 + O(e_n^5).
 \end{aligned} \tag{3.4}$$

Expanding $f(y_n)$ by Taylor's series about α , we get

$$f(y_n) = \frac{1}{4} \frac{c_3^2}{c_2} e_n^4 + \left(\frac{c_3 c_4}{c_2} - \frac{3}{4} \frac{c_3^3}{c_2^2} \right) e_n^5 + \left(\frac{3}{2} \frac{c_3 c_5}{c_2} - \frac{4 c_4 c_3^2}{c_2^2} + \frac{29}{16} \frac{c_3^4}{c_2^3} + \frac{c_4^2}{c_2} \right) e_n^6 + O(e_n^7). \tag{3.5}$$

Using (3.2), (3.3) and (3.5), the error tem becomes

$$e_{n+1} = \frac{1}{2} \frac{c_3}{c_2} e_n^2 + \left(\frac{c_4}{c_2} - \frac{7}{8} \frac{c_3^2}{c_2^2} \right) e_n^3 + \left(\frac{3}{2} \frac{c_5}{c_2} - \frac{3 c_3 c_4}{c_2^2} + \frac{27}{16} \frac{c_3^3}{c_2^3} \right) e_n^4 + O(e_n^5). \tag{3.6}$$

Now for Algorithm 2.3, we use (3.2) and (3.5) to compute z_n as follows

$$z_n = -2 \frac{f(y_n)}{f(x_n)} = -c_3^2 e_n^5 + \left(\frac{3}{2} \frac{c_3^3}{c_2} - 4 c_3 c_4 \right) e_n^6 + O(e_n^7). \tag{3.7}$$

Using (3.5) and (3.8), we have

$$y_n + z_n = \alpha + \frac{1}{2} \frac{c_3}{c_2} e_n^2 + \left(\frac{c_4}{c_2} - \frac{1}{4} \frac{c_3^2}{c_2^2} \right) e_n^3 + \left(\frac{3}{2} \frac{c_5}{c_2} - \frac{7}{4} \frac{c_3^3}{c_2^3} \right) e_n^4 + O(e_n^5). \tag{3.8}$$

Expanding $f(y_n + z_n)$ by Taylor's series about α , we obtain

$$f(y_n + z_n) = \frac{1}{4} \frac{c_3^2}{c_2} e_n^4 + \left(\frac{c_3 c_4}{c_2} - \frac{3}{4} \frac{c_3^3}{c_2^2} \right) e_n^5 + \left(\frac{3}{2} \frac{c_3 c_5}{c_2} - \frac{4 c_4 c_3^2}{c_2^2} + \frac{29}{16} \frac{c_3^4}{c_2^3} + \frac{c_4^2}{c_2} \right) e_n^6 + O(e_n^7). \tag{3.9}$$

Now using (3.2), (3.4), (3.7) and (3.9), we have the error tem as

$$e_{n+1} = \frac{1}{2} \frac{c_3}{c_2} e_n^2 + \left(\frac{c_4}{c_2} - \frac{3}{2} \frac{c_3^2}{c_2^2} \right) e_n^3 + \left(\frac{3}{2} \frac{c_5}{c_2} - \frac{11}{2} \frac{c_3 c_4}{c_2^2} + \frac{19}{4} \frac{c_3^3}{c_2^3} \right) e_n^4 + O(e_n^5) .. \tag{3.10}$$

The last equation shows that the convergence order of Algorithm 2.3 is 2. This completes the proof. □

4. Numerical Examples

In this section, we present some numerical examples to demonstrate the performance of the newly developed iterative method. In the following Tables 1-12, we compare our proposed methods (Algorithms 2.2 and 2.3) (NIM1) and (NIM2) with classical Newton’s method (NM), generalized Newton’s method (GNM), Chun et al. [6, Equations (35) and (36)] (BNM1) and (BNM2). All the computations for above mentioned methods are performed using software Maple and $\varepsilon = 10^{-10}$ as tolerance and also the following criteria is used for estimating the zero:

- (i) $\delta = |x_{n+1} - x_n| < \varepsilon$,
- (ii) $|f(x_n)| < \varepsilon$,
- (iii) Maximum numbers of iterations = 500.

Here N denotes the number of iterations.

Table 1. Comparison for $f_1(x) = (x - 2)^2$

$f(x)$	x_0	Method	N	$x [k]$	$f(x_n)$	δ
$f_1(x)$	1.97	NM	10	1.9999707031250000	$8.58e - 10$	$2.93e - 05$
		GNM	1	2.0000000000000000	$0.00e + 00$	$3.00e - 02$
		BNM1	1	2.0000000000000000	$0.00e + 00$	$3.00e - 02$
		BNM2	1	2.0000000000000000	$0.00e + 00$	$3.00e - 02$
		NIM1	1	2.0000000000000000	$0.00e + 00$	$3.00e - 02$
		NIM2	1	2.0000000000000000	$0.00e + 00$	$3.00e - 02$

Table 2. Comparison for $f_2(x) = \arctan^2 x$

$f(x)$	x_0	Method	N	$x [k]$	$f(x_n)$	δ
$f_2(x)$	0.02	NM	10	0.0000195243074874	$3.81e - 10$	$1.95e - 05$
		GNM	1	-0.0000053329067398	$2.84e - 11$	$2.00e - 02$
		BNM1	1	0.0000026713622895	$7.14e - 12$	$2.00e - 02$
		BNM2	1	0.0000026617546045	$7.08e - 12$	$2.00e - 02$
		NIM1	1	-0.0000053343294930	$2.85e - 11$	$2.00e - 02$
		NIM2	1	-0.0000053371757585	$2.85e - 11$	$2.00e - 02$

Table 3. Comparison for $f_3(x) = (e^x - 4x^2)^2$

$f(x)$	x_0	Method	N	$x [k]$	$f(x_n)$	δ
$f_3(x)$	-0.39	NM	12	-0.4077725133649210	$2.72e - 10$	$4.20e - 06$
		GNM	2	-0.4077767961647673	$1.16e - 13$	$3.05e - 04$
		BNM1	2	-0.4077767094044696	$1.78e - 27$	$1.69e - 05$
		BNM2	1	-0.4077777243328640	$1.59e - 11$	$1.78e - 02$
		NIM1	2	-0.4077767992980449	$1.25e - 13$	$3.10e - 04$
		NIM2	2	-0.4077768058455424	$1.43e - 13$	$3.22e - 04$

Table 4. Comparison for $f_4(x) = x^3 - x^2 - x + 1$

$f(x)$	x_0	Method	N	$x [k]$	$f(x_n)$	δ
$f_4(x)$	1.03	NM	11	1.0000148669755619	$4.42e - 10$	$1.49e - 05$
		GNM	2	1.0000000121033821	$2.93e - 16$	$2.20e - 04$
		BNM1	1	1.0000055621052384	$6.19e - 11$	$3.00e - 02$
		BNM2	1	1.0000009278935850	$1.72e - 12$	$3.00e - 02$
		NIM1	2	1.0000000119297000	$2.85e - 16$	$2.18e - 04$
		NIM2	2	1.0000000115885852	$2.69e - 16$	$2.15e - 04$

Table 5. Comparison for $f_5(x) = (\sin x - \cos x)^2$

$f(x)$	x_0	Method	N	$x [k]$	$f(x_n)$	δ
$f_5(x)$	0.7	NM	12	0.7853773818407633	$8.64e - 10$	$2.08e - 05$
		GNM	2	0.7853981633944398	$1.81e - 23$	$2.08e - 04$
		BNM1	2	0.7853981633972462	$8.17e - 26$	$1.07e - 04$
		BNM2	2	0.7853981633972653	$6.70e - 26$	$1.03e - 04$
		NIM1	2	0.7853981633944176	$1.84e - 23$	$2.09e - 04$
		NIM2	2	0.7853981633943728	$1.89e - 23$	$2.10e - 04$

Table 6. Comparison for $f_6(x) = (x^2 + \ln x^2 + \sin x^2)^2$

$f(x)$	x_0	Method	N	$x [k]$	$f(x_n)$	δ
$f_6(x)$.62	NM	13	0.6552172246879896	$5.83e - 10$	$4.35e - 06$
		GNM	2	0.6552215653222547	$9.30e - 16$	$1.98e - 04$
		BNM1	2	0.6552215708114534	$2.46e - 30$	$1.11e - 05$
		BNM2	1	0.6552215708114528	$2.35e - 29$	$1.48e - 05$
		NIM1	2	0.6552215653829742	$9.09e - 16$	$1.97e - 04$
		NIM2	2	0.6552215655027309	$8.69e - 16$	$1.95e - 04$

Table 7. Comparison for $f_7(x) = x^3 - 3x + 2$

$f(x)$	x_0	Method	N	$x [k]$	$f(x_n)$	δ
$f_7(x)$	1.06	NM	12	1.0000149394272540	$6.70e - 10$	$1.49e - 05$
		GNM	2	1.0000000565392868	$9.59e - 15$	$5.82e - 04$
		BNM1	2	1.0000000000000007	$1.51e - 30$	$1.94e - 05$
		BNM2	1	1.0000033833477819	$3.43e - 11$	$6.00e - 02$
		NIM1	2	1.0000000554730511	$9.23e - 15$	$5.77e - 04$
		NIM2	2	1.0000000533910456	$8.55e - 15$	$5.66e - 04$

Table 8. Comparison for $f_8(x) = (x - e^{\cos x})^2$

$f(x)$	x_0	Method	N	$x [k]$	$f(x_n)$	δ
$f_8(x)$	1.23	NM	13	1.3029548560201398	$4.26e - 10$	$9.15e - 06$
		GNM	2	1.3029638361281054	$1.39e - 13$	$9.27e - 04$
		BNM1	2	1.3029640012159444	$2.37e - 26$	$7.09e - 05$
		BNM2	2	1.3029640012160073	$1.40e - 28$	$4.04e - 05$
		NIM1	2	1.3029638401650150	$1.32e - 13$	$9.16e - 04$
		NIM2	2	1.3029638479906811	$1.20e - 13$	$8.94e - 04$

Table 9. Comparison for $f_9(x) = (x - \tan x)^2$

$f(x)$	x_o	Method	N	$x [k]$	$f(x_n)$	δ
$f_9(x)$	0.05	NM	1	0.0416722242070108	$5.83e - 10$	$8.33e - 03$
		GNM	1	0.0333444484140216	$1.53e - 10$	$1.67e - 02$
		BNM1	1	0.0288991908845572	$6.47e - 11$	$2.11e - 02$
		BNM2	1	0.0255627464167388	$3.10e - 11$	$2.44e - 02$
		NIM1	1	0.0318809320892903	$1.17e - 10$	$1.81e - 02$
		NIM2	1	0.0292994996253647	$7.03e - 11$	$2.07e - 02$

Table 10. Comparison for $f_{10}(x) = (x - \sin x)^2$

$f(x)$	x_o	Method	N	$x [k]$	$f(x_n)$	δ
$f_{10}(x)$	0.06	NM	1	0.0499987998456958	$4.34e - 10$	$1.00e - 02$
		GNM	1	0.0399975996913915	$1.14e - 10$	$2.00e - 02$
		BNM1	1	0.0346644425641644	$4.82e - 11$	$2.53e - 02$
		BNM2	1	0.0306651148409437	$2.31e - 11$	$2.93e - 02$
		NIM1	1	0.0382418400691131	$8.69e - 11$	$2.18e - 02$
		NIM2	1	0.0351448601057440	$5.23e - 11$	$2.49e - 02$

Table 11. Comparison for $f_{11}(x) = (x^2 + \sin \frac{x}{5} - \frac{1}{4})^2$

$f(x)$	x_o	Method	N	$x [k]$	$f(x_n)$	δ
$f_{11}(x)$	0.37	NM	11	0.4099740214194201	$3.36e - 10$	$1.80e - 05$
		GNM	2	0.4099948386255551	$8.27e - 12$	$1.70e - 03$
		BNM1	2	0.4099920179505115	$1.55e - 21$	$2.37e - 04$
		BNM2	2	0.4099920179891372	$2.43e - 33$	$3.20e - 05$
		NIM1	2	0.4099951109809942	$9.94e - 12$	$1.78e - 03$
		NIM2	2	0.4099957203043393	$1.42e - 11$	$1.95e - 03$

Table 12. Comparison for $f_{12}(x) = (x^2 + 7x - 30)^2$

$f(x)$	x_o	Method	N	$x [k]$	$f(x_n)$	δ
$f_{12}(x)$	3.03	NM	14	3.0000018395052141	$5.72e - 10$	$1.84e - 06$
		GNM	2	3.0000000003653009	$2.26e - 17$	$6.89e - 05$
		BNM1	1	3.0000004727268600	$3.78e - 11$	$3.00e - 02$
		BNM2	1	3.0000000032477361	$1.78e - 15$	$3.00e - 02$
		NIM1	2	3.0000000003636341	$2.23e - 17$	$6.88e - 05$
		NIM2	2	3.0000000003603197	$2.19e - 17$	$6.84e - 05$

5. Conclusions

In the present work, we have proposed two new iterative methods (NIM1) and (NIM2) with convergence order 2 for finding the multiple roots of nonlinear equations. The numerical results presented in the Tables 1-12 given in the previous section reveal that our iterative methods (NIM1) and (NIM2) are even comparable with the methods developed by Chun et al. [6] (BNM1) and

(BNM2) with convergence order 3. The idea and technique employed in this paper can be developed to higher-order multi-step iterative methods for solving nonlinear equations having multiple roots.

References

- [1] S. Abbasbandy, Improving Newton-Raphson method for nonlinear equations modified Adomian decomposition method, *Appl. Math. Comput.*, **145** (2003), 887-893. doi: 10.1016/S0096-3003(03)00282-0
- [2] G. Adomian, *Nonlinear Stochastic Systems and Applications to Physics*, Kluwer Academic Publishers, Dordrecht, 1989.
- [3] E. Babolian, J. Biazar, On the order of convergence of Adomian method, *Appl. Math. Comput.*, **130** (2002), 383-387. doi: 10.1016/S0096-3003(01)00103-5
- [4] E. Babolian, J. Biazar, Solution of nonlinear equations by modified Adomian decomposition method, *Appl. Math. Comput.*, **132** (2002), 167-172. doi: 10.1016/S0096-3003(01)00184-9
- [5] C. Chun, Iterative methods improving Newton's method by the decomposition method, *Comput. Math. Appl.*, **50** (2005), 1559-1568. doi: 10.1016/j.camwa.2005.08.022
- [6] C. Chun, H.J. Bae, B. Neta, New families of nonlinear third-order solvers for finding multiple roots, *Comput. Math. Appl.*, **57** (2009), 1574-1582. doi: 10.1016/j.camwa.2008.10.070
- [7] C. Chun, Y. Ham, A one-parameter fourth-order family of iterative methods for nonlinear equations, *Appl. Math. Comput.*, **189** (2007), 610-614. doi: 10.1016/j.amc.2006.11.113
- [8] C. Chun, B. Neta, A third-order modification of Newton's method for multiple roots, *Appl. Math. Comput.*, **211** (2009), 474-479. doi: 10.1016/j.amc.2009.01.087
- [9] C. Chun, B. Neta, Basin of attraction for Zhou-Chen-Song fourth order family of methods for multiple roots, *Math. Comput. Simulation*, **109** (2015), 74-91. doi: 10.1016/j.matcom.2014.08.005
- [10] V. Daftardar-Gejji, H. Jafari, An iterative method for solving nonlinear functional equations, *J. Math. Anal. Appl.*, **316** (2006), 753-763. doi: 10.1016/j.jmaa.2005.05.009
- [11] V.I. Hasanov, I.G. Ivanov, G. Nedzhibov, A new modification of Newton's method, *Appl. Math. Engin. Econom. (Sozopol, 2001)*, pp. 278-286, Heron Press, Sofia, 2002
- [12] H.H.H. Homeier, On Newton-type methods for multiple roots with cubic convergence, *J. Comput. Appl. Math.*, **231** (2009), 249-254. doi: 10.1016/j.cam.2009.02.006
- [13] E. Isaacson, H.B. Keller, *Analysis of Numerical Methods*, John Wiley & Sons, Inc., New York, 1966.
- [14] B. Neta, C. Chun, On a family of Laguerre methods to find multiple roots of nonlinear equations, *Appl. Math. Comput.*, **219** (2013), 10987-11004. doi: 10.1016/j.amc.2013.05.002
- [15] B. Neta, C. Chun, M. Scott, On a development of iterative methods for multiple roots, *Appl. Math. Comput.*, **224** (2013), 358-361. doi: 10.1016/j.amc.2013.08.077

- [16] M.A. Noor, K.H. Noor, Some iterative schemes for nonlinear equations, *Appl. Math. Comput.*, **183** (2006), 774-779. **doi:** 10.1016/j.amc.2006.05.084
- [17] N. Osada, An optimal multiple root-finding method of order three, *J. Comput. Appl. Math.*, **51** (1994), 131-133. **doi:** 10.1016/0377-0427(94)00044-1
- [18] E. Schröder, Über unendlich viele algorithmen zur auflösung der gleichungen, *Math. Ann.*, **2** (1870), 317-365. **doi:** 10.1007/BF01444024.