Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080532

A Family of Iterative Schemes for Finding Zeros of Nonlinear Equations having Unknown Multiplicity

Muhammad Aslam Noor[1] *and Farooq Ahmed Shah*[2,*]

[1] Department of Mathematics, COMSATS Institute of Information Technology, Park Road, Islamabad, Pakistan
[2] Department of Mathematics, COMSATS Institute of Information Technology, Kamra Road, Attock Campus, Pakistan

Received: 4 Sep. 2013, Revised: 2 Dec. 2013, Accepted: 3 Dec. 2013
Published online: 1 Sep. 2014

Abstract: In this paper, we suggest and analyze a new family of iterative methods for finding zeros of multiplicity of nonlinear equations by using the variational iteration technique. These new methods include the Halley method and its variants forms as special cases. We also give several examples to illustrate the efficiency of these methods. Comparison with modified Newton method is also given. These new methods can be considered as an alternative to the modified Newton method. This technique can be used to suggest a wide class of new iterative methods for solving system of nonlinear equations.

Keywords: Variational iteration technique, Zeros of multiplicity, Newton method, Iterative methods, Convergence, Examples

1 Introduction

One of the most important and challenging problems in scientific and engineering applications is to find the solution of the nonlinear equations $f(x) = 0$. Various iterative methods are being developed for finding the simple roots of the nonlinear equation $f(x) = 0$, by using several different techniques such as Taylor series, quadrature formulas, homotopy and decomposition methods, see [1,2,3,4,5,6,7,8,9,10,11,13,12,14,15]. Some time we come across the nonlinear equations which have zeros of multiplicity $m \geq 2$. The methods derived for finding simple roots can not be applied for finding zeros of multiplicity of the nonlinear equations.

The aim of the present paper is to extend the variational iteration technique for finding zeros of multiplicity of the nonlinear equations. We can classify the problem of finding multiple roots into two types. First, multiplicity is known and secondly, multiplicity is unknown.

If the multiplicity of the root is greater than one, then the Newton method

$$x_{n+1} = x_n - \frac{f(x_n)}{\acute{f}(x_n)}, \qquad n = 0, 1, 2, \cdots,$$

generates the sequence of iterations that converge to the root linearly and some time it diverges. In order to overcome this drawback of restricted convergence and to preserve the order of convergence, the Newton method is modified by using the knowledge of multiplicity $m \geq 2$, for finding multiple roots, see [1,14].

$$x_{n+1} = x_n - m\frac{f(x_n)}{\acute{f}(x_n)}, \qquad n = 0, 1, 2, \cdots.$$

For finding multiple roots, iterative methods for the first case are developed [2,5,6,7]. For second case, when multiplicity is not known, the Newton method is modified as:

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{f'(x_n)^2 - f(x_n)f''(x_n)}, \qquad n = 0, 1, 2, \cdots,$$

which preserves the second order convergence for finding multiple roots, see [1,14].

Variational iteration technique was effectively used by Noor [8] and some efficient iterative methods were suggested which can be considered as good alternate of the Newton method. Noor and Shah [10,11] also suggested some higher order iterative methods for solving nonlinear equations for finding simple roots and for

* Corresponding author e-mail: farooqhamdani@gmail.com

finding zeros of multiplicity of the nonlinear equations. This technique is recently extended for systems of nonlinear equations [12]. We observe that this technique not only plays an important role for the solution of nonlinear equation for simple roots as well as for multiple roots. In this paper, we use the variational iteration technique to suggest and analyze some modifications of Newton method for finding the multiple roots of the nonlinear equations having unknown multiplicity.

2 Construction of iterative methods

In this section, we describe the variational iteration technique for obtaining the main recurrence relations. These relations generate the iterative methods for finding multiple roots of nonlinear equations. We consider the nonlinear equation having zeros of multiplicity $m \geq 2$, as:

$$f(x) = 0, \tag{1}$$

which can be written in the following equivalent form as:

$$x = H(x). \tag{2}$$

We consider the auxiliary function $H(x)$ defined by

$$H(x) = x + \lambda \left[\frac{f(x)}{f'(x)}\right] g(x), \tag{3}$$

where $g(x)$, is the auxiliary function and λ is the Lagrange multiplier which can be identified by using optimality criteria.

We have to find the methods for finding multiple roots of nonlinear equations with unknown multiplicity. The ratio of $f(x)$ and its derivative $f'(x)$ which is involved in (3). This ratio essentially remove the multiplicity of the roots whatever it has before the implementation of the method. This invisible elimination of the unknown multiplicity makes it easy to employ the methods for further process in finding multiple roots.

Using the optimality criteria, we obtain the value of λ as:

$$\lambda = -\frac{[f'(x)]^2}{([f'(x)]^2 - f(x)f''(x))g(x) + f(x)f'(x)g'(x)}. \tag{4}$$

From (3) and (4), we obtain

$$H(x) = x - \frac{f(x)f'(x)g(x)}{([f'(x)]^2 - f(x)f''(x))g(x) + f(x)f'(x)g'(x)}. \tag{5}$$

Now combining (2) and (5), we obtain

$$x = H(x) = x - \frac{f(x)f'(x)g(x)}{([f'(x)]^2 - f(x)f''(x))g(x) + f(x)f'(x)g'(x)}. \tag{6}$$

This fixed point formulation enables us to suggest the following iterative scheme as:

Algorithm 2.1. For a given x_0, find the approximation solution x_{n+1} by the following iterative scheme:

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)g(x_n)}{([f'(x_n)]^2 - f(x_n)f''(x_n))g(x_n) + f(x_n)f'(x_n)g'(x_n)},$$

which is the main iteration scheme for finding multiple roots of nonlinear equations. We now discuss the following some special cases.

Case I. Let $g(x) = e^{-\alpha x_n}$. Then from Algorithm 2.1, we obtain the following iterative method for solving the nonlinear equations having unknown zeros of multiplicity.

Algorithm 2.2. For a given x_0, find the approximation solution x_{n+1} by the following iterative scheme:

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{([f'(x_n)]^2 - f(x_n)f''(x_n)) - \alpha f(x_n)f'(x_n)}.$$

If $\alpha = 0$, then Algorithm 2.2 reduces to the well known modified Newton method [1,14].

If $\alpha = -\frac{f''(x_n)}{2f'(x_n)}$, then Algorithm 2.2 reduces to the following third-order convergent iterative method for finding simple root of nonlinear equations.

Algorithm 2.3. For a given x_0, find the approximation solution x_{n+1} by the following iterative scheme:

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2[f'(x_n)]^2 - f(x_n)f''(x_n)},$$

which is well known Halley method [1,8,14] and has third order convergence.

Case II. Let $g(x) = e^{-\alpha f(x_n)}$. Then, from Algorithm 2.1, we obtain the following iterative method for solving the nonlinear equations having unknown zeros of multiplicity.

Algorithm 2.4. For a given x_0, find the approximation solution x_{n+1} by the following iterative scheme:

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{([f'(x_n)]^2 - f(x_n)f''(x_n)) - \alpha f(x_n)f'(x_n)^2}.$$

If $\alpha = 0$, then Algorithm 2.4 reduces to the well known modified Newton method [1,14].

If $\alpha = -\frac{f''(x_n)}{2f'(x_n)^2}$, then Algorithm 2.4 reduces to the Algorithm 2.3.

Case III. Let $g(x) = e^{\frac{\alpha}{f'(x_n)}}$. Then, from Algorithm 2.1, we obtain the following iterative method for solving the nonlinear equations having unknown zeros of multiplicity.

Algorithm 2.5. For a given x_0, find the approximation solution x_{n+1} by the following iterative scheme:

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)^2}{([f'(x_n)]^3 - f(x_n)f''(x_n)f'(x_n) - \alpha f(x_n)f''(x_n)}.$$

If $\alpha = 0$, then Algorithm 2.5 reduces to the well known modified Newton method[1,14].

If $\alpha = -\frac{f'(x_n)}{2}$, then Algorithm 2.5 reduces to the Algorithm 2.3, the well known Halley method[1,8,14].

Case IV. Let $g(x) = e^{-\alpha\frac{f(x_n)}{f'(x_n)}}$. Then, from Algorithm 2.1, we obtain the following iterative method for solving the nonlinear equations having unknown zeros of multiplicity.

Algorithm 2.6. For a given x_0, find the approximation solution x_{n+1} by the following iterative scheme:

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)^2}{[(f'(x_n)^2 - f(x_n)f''(x_n))]f'(x_n) - \alpha f(x_n)}.$$

If $\alpha = 0$, then Algorithm 2.6 reduces to the well known modified Newton method [1,14].

If $\alpha = \frac{f''(x_n)}{2f'(x_n)}$, then Algorithm 2.6 reduces to the Algorithm 2.3.

Remark 2.1. It should be noted that never decide to select such a value of the parameter α, which makes the denominator zero in the derived methods.

Remark 2.2. Sign of α, should be selected so as to make the denominator largest in magnitude in above methods to obtain the good results.

3 Convergence analysis

In this section, we consider the convergence criteria of the main iterative scheme defined above as Algorithm 2.1 developed in this paper by using the fixed point method.

Theorem 1. *Let r be a multiple root of unknown multiplicity m of a sufficiently differentiable function $f : \mathscr{D} \subset \mathbb{R} \to \mathbb{R}$ for an open interval in $\mathscr{D}$. If x_0, is in the neighborhood of r, then for any auxiliary function $g(x)$ the main recurrence scheme defined as Algorithm 2.1 and consequently, all the Algorithms derived from this relation are at least quadratically convergent.*

Proof. Here we study the function (5) associated with Algorithm 2.1 of the type

$$H(x) = x - \frac{f(x)f'(x)g(x)}{([f'(x)]^2 - f(x)f''(x))g(x) + f(x)f'(x)g'(x)},$$

where

$$f(x_n) = (x-r)^m h(x) \quad (7)$$

$$f'(x_n) = m(x-r)^{m-1}h(x) + (x-r)^m h'(x) \quad (8)$$

and

$$f''(x_n) = m(m-1)(x-r)^{m-2}h(x)$$

$$+2m(x-r)^{m-1}h'(x) + (x-r)^m h''(x) \quad (9)$$

It is assumed that r is an m-fold root of $f(x)$ and $h(r) \neq 0$.

Now, replacing (7), (8) and (9) in (5), and after simplifying, we get

$$H(r) = r \quad (10)$$

and also with the help of computer program, differentiating $H(x)$ with respect to x and simple manipulations yield that

$$H'(r) = 0, \quad (11)$$

and

$$H''(r) = 2\left[\frac{-h'(r)g(r) + mg'(r)h(r)}{mh(r)g(r)}\right] \neq 0. \quad (12)$$

This shows that the Algorithm 2.1 has at least second order convergence and consequently, all the Algorithms derived from this relation are at least quadratically convergent.

Remark 3.1. In previous section, third order convergent methods are derived for obtaining the simple roots of nonlinear equations due to some particular value of involved in the Algorithms. New proposed methods are Halley method and its variants which is the novelty of this technique.

4 Numerical results

We now present some examples to illustrate the efficiency of the new developed two-step iterative methods (see Tables 4.1-4.12). We compare the Modified Newton method (MNM)[1,14], Algorithm 2.2, Algorithm 2.4, Algorithm 2.5 and Algorithm 2.6, which are introduced here in this paper. All computations are done using the MAPLE using 60 digits floating point arithmetics (Digits: =60). We will use $\varepsilon = 10^{-32}$. The following stopping criteria are used for computer programs.

$$(i)\ |x_{n+1} - x_n| \leq \varepsilon,\ (ii)\ |f(x_n)| \leq \varepsilon.$$

The computational order of convergence p approximated for all the examples in Tables 4.1-4.12, (see [15]) by means of

$$p = \frac{ln(|x_{n+1} - x_n|/|x_n - x_{n-1}|)}{ln(|x_n - x_{n-1}|/|x_{n-1} - x_{n-2}|)}.$$

Example 4.1. We consider the nonlinear equation

$$f(x) = x^4 - 2x^2 + 1.$$

We consider $\alpha = 1$ and $\alpha = 0.5$ for all the methods to compare the numerical results in Table 4.1 and Table 4.2 respectively.

Table 4.1

Method	IT	x_n	$\|f(x_n)\|$	δ	p
MNM	11	1.0000	0.00e-01	2.33e-15	2.398
Alg 2.2	5	1.0000	0.00e-01	3.68e-12	2.953
Alg 2.4	8	1.0000	0.00e-01	1.99e-16	1.999
Alg 2.5	9	1.0000	0.00e-01	3.94e-17	2.035
Alg 2.6	6	1.0000	0.00e-01	2.31e-18	2.390

Table 4.1 depicts the numerical results of example 4.1. We use the initial guess $x_0 = 0.5$ for the computer program for $\alpha = 1$.

Table 4.2

Method	IT	x_n	$\|f(x_n)\|$	δ	p
MNM	11	1.0000	0.00e-01	2.33e-15	2.398
Alg 2.2	4	1.0000	0.00e-01	1.68e-08	2.699
Alg 2.4	6	1.0000	0.00e-01	2.09e-11	2.1680
Alg 2.5	5	1.0000	0.00e-01	9.51e-12	2.034
Alg 2.6	6	1.0000	0.00e-01	2.40e-19	2.006

Table 4.2 depicts the numerical results of example 4.1. We use the initial guess $x_0 = 0.5$, for the computer program for $\alpha = 0.5$.

Example 4.2. We consider the nonlinear equation

$$f_2(x) = x^{12} - 2x^6 + 1.$$

We consider $\alpha = 1$ and $\alpha = 0.5$ for all the methods to compare the numerical results in Table 4.3 and Table 4.4 respectively.

Table 4.3

Method	IT	x_n	$\|f(x_n)\|$	δ	p
MNM	14	1.0000	0.00e-01	1.09e-13	2.0135
Alg 2.2	13	1.0000	0.00e-01	4.13e-14	2.2094
Alg 2.4	12	1.0000	0.00e-01	4.17e-10	1.9995
Alg 2.5	12	1.0000	0.00e-01	4.17e-10	2.1995
Alg 2.6	12	1.0000	0.00e-01	4.14e-10	2.0995

Table 4.3 shows the efficiency of the methods for example 4.2. We use the initial guess $x_0 = 0.2$, for the computer program for $\alpha = 1$. Number of iterations and computational order of convergence gives us an idea about the better performance of the new methods.

Table 4.4

Method	IT	x_n	$\|f(x_n)\|$	δ	p
MNM	14	1.0000	0.00e-01	1.09e-13	2.0135
Alg 2.2	13	1.0000	0.00e-01	6.09e-10	2.0660
Alg 2.4	13	1.0000	0.00e-01	4.17e-10	2.0995
Alg 2.5	14	1.0000	0.00e-01	4.17e-10	1.9995
Alg 2.6	13	1.0000	0.00e-01	4.17e-10	1.6995

Table 4.4 shows the efficiency of the methods for example 4.2. We use the initial guess $x_0 = 0.2$, for the computer program for $\alpha = 0.5$. Number of iterations and computational order of convergence gives us an idea about the better performance of the new methods.

Example 4.3. We consider the nonlinear equation

$$f_3(x) = -8 + 36x - 90x^2 + x^9 - 9x^8 + 36x^7$$
$$+147x^3 - 87x^6 + 144x^5 - 171x^4.$$

We consider $\alpha = 1$ and $\alpha = 0.5$ for all the methods to compare the numerical results in Table 4.5 and Table 4.6 respectively.

Table 4.5

Method	IT	x_n	$\|f(x_n)\|$	δ	p
MNM	7	1.9999	0.00e-01	6.64e-12	2.0064
Alg 2.2	7	1.9999	0.00e-01	2.44e-10	2.9403
Alg 2.4	5	1.9999	0.00e-01	2.49e-05	2.8977
Alg 2.5	7	1.9999	0.00e-01	4.17e-10	2.2995
Alg 2.6	7	1.9999	0.00e-01	8.66e-13	1.9880

In Table 4.5, the numerical results for example 4.3 are described. We use the initial guess $x_0 = 1.5$ for the computer program for $\alpha = 1$. We observe that all the methods approach to the approximate solution after equal number of iterations but the computational order of convergence has little bit difference.

Table 4.6

Method	IT	x_n	$\|f(x_n)\|$	δ	p
MNM	7	1.9999	0.00e-01	6.64e-12	2.0064
Alg 2.2	6	1.9999	0.00e-01	1.25e-06	2.1777
Alg 2.4	6	1.9999	0.00e-01	1.64e-06	2.5298
Alg 2.5	6	1.9999	0.00e-01	2.17e-10	2.1995
Alg 2.6	6	1.9999	0.00e-01	1.94e-11	2.0456

In Table 4.6, the numerical results for example 4.3 are described. We use the initial guess $x_0 = 1.5$ for the computer program for $\alpha = 0.5$.. We observe that all the methods approach to the approximate solution after equal number of iterations but the computational order of convergence has little bit difference.

Example 4.4. We consider the nonlinear equation

$$f_4(x) = x^{10} - \frac{30}{31}x^6 - 2x^5 + \frac{225}{961}x^2 + \frac{30}{31}x + 1.$$

We consider $\alpha = 1$ and $\alpha = 0.5$ for all the methods to compare the numerical results in Table 4.7 and Table 4.8 respectively.

Table 4.7

Method	IT	x_n	$\|f(x_n)\|$	δ	p
MNM	7	1.08828	0.00e-01	2.21e-18	1.9241
Alg 2.2	6	1.08828	0.00e-01	1.73e-20	2.0010
Alg 2.4	5	1.08828	0.00e-01	1.86e-11	2.2158
Alg 2.5	7	1.08828	0.00e-01	1.54e-14	2.0976
Alg 2.6	5	1.08828	0.00e-01	1.38e-16	2.0033

Table 4.7 shows the numerical results for example 4.4. For the computer program we use the initial guess $x_0 = 1$, and $\alpha = 1$. We note that the new derived methods have better computational order of convergence and approach to the desired result in less number of iterations.

Table 4.8

Method	IT	x_n	$\|f(x_n)\|$	δ	p
MNM	7	1.08828	0.00e-01	2.21e-18	1.9241
Alg 2.2	6	1.08828	0.00e-01	6.68e-20	2.0005
Alg 2.4	5	1.08828	0.00e-01	2.23e-20	2.0403
Alg 2.5	7	1.08828	0.00e-01	5.53e-19	2.1859
Alg 2.6	5	1.08828	0.00e-01	1.90e-18	2.0002

Table 4.8 shows the numerical results for example 4.4. For the computer program we use the initial guess $x_0 = 1$, and $\alpha = 0.5$. We note that the new derived methods have better computational order of convergence and approach to the desired result in less number of iterations.

Example 4.5. We consider the nonlinear equation

$$f_5(x) = x^6 - 2x^5 + \frac{5}{3}x^4 - \frac{74}{27}x^3 + \frac{59}{27}x^2 - \frac{56}{81}x + \frac{784}{729}.$$

We consider $\alpha = 1$ and $\alpha = 0.5$ for all the methods to compare the numerical results in Table 4.9 and Table 4.10 respectively.

Table 4.9

Method	IT	x_n	$\|f(x_n)\|$	δ	p
MNM	8	1.3333	0.00e-01	3.64e-16	2.0257
Alg 2.2	5	1.3333	0.00e-01	2.54e-21	2.0255
Alg 2.4	6	1.3333	0.00e-01	8.82e-09	2.3741
Alg 2.5	7	1.3333	0.00e-01	1.25e-16	1.9993
Alg 2.6	5	1.3333	0.00e-01	8.69e-13	2.0277

In Table 4.9, we show the numerical results for the example 4.5. We use the initial guess $x_0 = 1$, and $\alpha = 1$, for the computer program. We observe that the new methods approach to the desired approximate solution in equal or less number of iterations. We calculate the computational order of convergence for all the methods which verify the rate of convergence and efficiency of the methods.

Table 4.10

Method	IT	x_n	$\lvert f(x_n)\rvert$	δ	p
MNM	8	1.3333	0.00e-01	3.64e-16	2.0257
Alg 2.2	8	1.3333	0.00e-01	1.13e-18	2.1871
Alg 2.4	5	1.3333	0.00e-01	1.61e-10	1.8454
Alg 2.5	8	1.3333	0.00e-01	1.50e-18	2.0046
Alg 2.6	7	1.3333	0.00e-01	1.40e-17	1.9998

In Table 4.10, we show the numerical results for the example 4.5. We use the initial guess $x_0 = 1$, and $\alpha = 0.5$, for the computer program. We observe that the new methods approach to the desired approximate solution in equal or less number of iterations. We calculate the computational order of convergence for all the methods which verify the rate of convergence and efficiency of the methods.

Example 4.6. We consider the nonlinear equation

$$f_6(x) = \left(x^3 - \frac{3}{4}\right)^3.$$

We consider $\alpha = 1$ and $\alpha = 0.5$ for all the methods to compare the numerical results in Table 4.11 and Table 4.12 respectively.

Table 4.11

Method	IT	x_n	$\lvert f(x_n)\rvert$	δ	p
MNM	6	0.9085	1.62e-56	3.05e-10	2.0616
Alg 2.2	5	0.9085	6.60e-34	1.29e-06	2.1494
Alg 2.4	5	0.9085	3.77e-44	3.51e-08	2.2385
Alg 2.5	5	0.9085	2.45e-47	1.03e-08	2.1434
Alg 2.6	5	0.9085	3.49e-44	4.15e-08	2.1550

In Table 4.11, we show the numerical results for the example 4.6. We use the initial guess $x_0 = 0.5$ and $\alpha = 1$. for the computer program. We observe that the new methods approach to the desired approximate solution in less number of iterations.

Table 4.12

Method	IT	x_n	$\lvert f(x_n)\rvert$	δ	p
MNM	6	0.9085	1.62e-56	3.05e-10	2.06156
Alg 2.2	5	0.9085	1.62e-38	4.12e-07	2.43355
Alg 2.4	5	0.9085	1.66e-64	1.42e-11	2.02849
Alg 2.5	5	0.9085	2.25e-62	3.22e-11	2.04121
Alg 2.6	5	0.9085	8.41e-47	1.18e-08	2.11010

In Table 4.12, we show the numerical results for the example 4.6. We use the initial guess $x_0 = 0.5$ and $\alpha = 0.5$. for the computer program. We observe that the new methods perform in better way and approach to the desired approximate solution in less number of iterations.

5 Conclusion

In this paper, we have suggested some new iterative methods for obtaining multiple roots of nonlinear equations by using variational iteration technique. The suggested methods have significance that these methods can be applied when multiplicity of the root is not known. From the numerical examples, it is clear that all the methods introduced in this paper perform better than the modified Newton method for obtaining multiple roots of nonlinear equations. Using the technique and idea, one can suggest and analyze higher order iterative methods for solving nonlinear equations as well as system of nonlinear equations. We would like to point out that Halley method and its variant forms are acquired from these methods by suitable and appropriate selection of the parameter α, which is also the innovative and narrative aim of the presented technique.

Acknowledgement

The authors are grateful to Dr. S. M. Junaid Zaidi, Rector COMSATS Institute of information Technology, Islamabad, Pakistan for providing excellent research environment and facilities. The authors are also grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] R. L. Burden and J. D. Faires, Numerical Analysis, PWS Publishing Company, Bostan, USA, (2001).
[2] C. Chun, H. J. Bae and B. Neta, Computers and Mathematics Applications, **57**, 1574–1582 (2009).
[3] E. Hansen and M. Patrick, Numerische Mathematik, **27**, 257–259 (1977).

[4] J. H. He, Journal of Computational and Applied Mathematics, **207**, 3-17 (2007) .
[5] H.H.H. Homeier, Journal of Computational and Applied Mathematics, **231**, 249 –254 (2009).
[6] B. Jovanovic, Mat. Vesnik., **9**, 365–369 (1972).
[7] B. Neta and A. N. Johnson, Computers and Mathematics Applications, **(55)**, 2012–2017 (2008).
[8] M. A. Noor, Applied Mathematics and Computation, **190**, 553-558 (2007).
[9] M. A. Noor, Applied Mathematics and Computation, **191**, 128–131 (2007).
[10] M. A. Noor and F. A. Shah, Journal of Applied Mathematics and Computation, **31**, 247–254 (2009).
[11] M. A. Noor, F. A. Shah, K. I. Noor and E. Al-said, Scientific Research and Essays, **6**, 1344–1350 (2011).
[12] M.A. Noor, M. Waseem, K. I. Noor and E. Al-Said, Optimization Letters, DOI: 10.1007/s11590-012-0479-3 (2012).
[13] N. Osada, Journal of Computational and Applied Mathematics, **98**, 311-315 (1998).
[14] J. F. Traub, Iterative methods for solutions of equations, Printice Hall, New York (1964).
[15] S. Weerakoon and T.G.I. Fernando, Applied Mathematics Letters, **13**, 87–93 (2000).

M. Aslam Noor earned his PhD degree from Brunel University, London, UK (1975) in the field of Numerical Analysis and Optimization (Applied Mathematics). He has a vast experience of teaching and research at university levels in various countries including Pakistan, Iran, Canada, Saudi Arabia and UAE. He has more than 750 research papers to his credit which are published in leading world class journals. He is versatile in his area of interest and specialization which covers many dimensions of Mathematical and Engineering sciences. A few of those are Variational Inequalities, Optimization, Operations Research, Numerical Analysis, Equilibrium and Economics with applications in Industry, Neural and Biosciences. His research contains a large number of innovative ideas and techniques. In 2009, one of his papers won the 'Best Research Paper award' by HEC, Pakistan. He has successfully supervised several PhD and MS students. He was titled as the 'Top Mathematician of the Muslim World' in 2008, by Organization of Islamic Countries (OIC). He has been ranked as one of the leading Scientists and Engineers in Pakistan. He was honored by the President of Pakistan with the 'President's Award for pride of performance' on August 14, 2008, for his extraordinary contributions in the field of Mathematical Sciences. Currently, He is a member of the Editorial Board of several reputed international journals of Mathematics and Engineering sciences. He has participated in several international conferences as invited speaker.

F. Ahmed Shah Assistant Professor Department of Mathematics (CIIT ATTOCK) earned his PhD degree from COMSATS Institute of Information Technology, Islamabad in the field of Numerical Analysis and Optimization (Applied Mathematics). His area of interest in Numerical Analysis is iterative methods for solving linear and nonlinear equations. He has more than 10 years of teaching and research experience. He has not only published research articles in reputed international journals of mathematical sciences but he is a reviewer too. He is also an active member of many mathematical societies.