Variational iteration technique for finding multiple roots of nonlinear equations

Muhammad Aslam Noor\textsuperscript{1,2,*}, Farooq Ahmed Shah\textsuperscript{1}, Khalida Inayat Noor\textsuperscript{1} and Eisa Al-Said\textsuperscript{2}

\textsuperscript{1}Department of Mathematics, Comsats Institute of Information Technology, Park Road, Islamabad, Pakistan.
\textsuperscript{2}Mathematics Department, College of Science, King Saud University, Riyadh 11451, Saudi Arabia.

Accepted 23 February, 2011

In this paper, we suggest and analyze some new iterative methods for finding multiple roots of nonlinear equations by using the variational iteration technique. The technique generates the methods of higher order. Here, the new methods are of second and third order convergence. We also give several examples to illustrate the efficiency of these methods. Comparison with other similar methods is also given.

Key words: Variational iteration, multiple roots, Newton method, Iterative methods.

INTRODUCTION

One of the most important and challenging problems in scientific and engineering applications is to find the solution of the nonlinear equations. In recent years, several numerical methods for finding simple zeroes of nonlinear equation \( f(x) = 0 \) have been developed by using several different techniques including Taylor series, quadrature formulas, homotopy and decomposition methods by Noor and Noor, (2007, a, b), Noor (2007), Noor and Shah (2009). For finding the multiple roots of nonlinear equations \( f(x) = 0 \), some iterative methods are developed by Chun et al. (2009, a) Hansen et al. (1977), Homeier (2009) and Osada (1994, 1998).

Inspired and motivated by the research going on in this field, we develop some iterative method for finding the multiple roots of nonlinear equation \( f(x) = 0 \) having multiplicity \( m \), that is, \( f^{(j)}(\alpha) = 0, j = 0, 1, \ldots, m-1 \) and \( f^{(m)}(\alpha) \neq 0 \), where \( \alpha \) is the root. We use the variational iterative method for suggesting some new iterative methods for finding the multiple roots of the nonlinear equations \( f(x) = 0 \).

It is worth mentioning that the variational iterative method was introduced by Inokuti et al. (1978). However, it was He (1999, 2007), who used this technique for finding the approximate solutions of linear and nonlinear problems. Noor (2007) and Noor et al. (2009) used the variational iteration method for suggesting a wide class of iterative methods for solving the nonlinear equations \( f(x) = 0 \). We show that the variational iteration method can be used to find the approximate multiple roots of the nonlinear equations having multiplicity \( m \). We also discuss several special cases of our results. To be more precise, we recall the well known modified Newton method for finding multiple roots of the nonlinear equations \( f(x) = 0 \) is given by:

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

which converges quadratically (Traub, 1964), where the zero of \( f(x) \) has multiplicity \( m \). There exist some known cubic convergent methods such as Halley method. Hansen and Patrick (1977) extended the Halley method for finding multiple roots of the nonlinear equations

\textsuperscript{*}Corresponding author. E-mail: mnoor.c@ksu.edu.sa, noormaslam@hotmail.com.

2000 AMS Classification: 65Bxx.
\[ f(x) = 0 \], which is given by:

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

The third order Osada's method (1994), for obtaining multiple roots of nonlinear equations is:

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

and Euler-Chebshev third-order multiple roots finding method is given by Traub (1964) as:

\[ x_{n+1} = x_n - \frac{m(3-m)}{2} \frac{f(x_n)}{f'(x_n)} - \frac{m^2}{2} \frac{f(x_n)^2}{f''(x_n)}. \] (4)

He (1999, 2007) developed the variational iteration technique and was used by Noor (2007) and Noor and Shah (2009) to develop some efficient iterative methods for solving nonlinear equations. In this paper, we show that the variational iterative method can be extended for developing some new iterative method of higher order for finding the multiple roots of the nonlinear equations. Several examples are given to illustrate the efficiency and the implementation of these new iterative methods. The results obtained in this may stimulate further research area in this fast growing and dynamic field.

CONSTRUCTION OF ITERATIVE METHODS

In this study, using the variational iteration technique, we suggest some iterative methods for finding the multiple roots of the nonlinear equations. For this purpose, we consider \( \phi(x_n) \) an iteration auxiliary function of order \( p \geq 1 \), \( m \) is the multiplicity of the zero of \( f(x) \) and \( \lambda \) is unknown Lagrange's multiplier such that:

\[ x = \phi(x) + \lambda [f(x)g(x)]^m. \] (5)

This fixed point function enables to suggest the following scheme for finding the approximate solution of \( f(x) = 0 \) by:

\[ x_{n+1} = \phi(x_n) + \lambda [f(x_n)g(x_n)]^m. \] (6)

Using the optimality condition, we have:

\[ \lambda = -\frac{m\phi'(x_n)}{p[f(x_n)g(x_n)]^m - [f'(x_n)g(x_n) + f(x_n)g'(x_n)]}. \] (7)

From (6) and (7), we have:

\[ x_{n+1} = \phi(x_n) - \frac{m\phi'(x_n)f(x_n)g(x_n)}{p[f'(x_n)g(x_n) + f(x_n)g'(x_n)]}. \] (8)

This is a recurrence relation which generates iterative methods of order \( p + 1 \) for solving the nonlinear equation \( f(x) = 0 \). We use (8) to suggest several iterative methods of higher order for finding multiple roots of nonlinear equation and this is the main motivation. We now discuss some special cases.

Case 1

In this case, we will modify (8) for obtaining iterative methods of second order for solving nonlinear equations.

Let \( \phi(x) = I \) and \( p = 1 \).

Then relation (8) becomes:

\[ x_{n+1} = x_n - \frac{m f(x_n)g(x_n)}{f'(x_n)g(x_n) + f(x_n)g'(x_n)}. \] (9)

This is the iteration relation which generates at least second order convergent methods for finding multiple roots of the nonlinear equation. Now we consider some special values of \( g(x_n) \) to suggest some iterative methods for solving nonlinear equation \( f(x) = 0 \).

i) Let \( g(x_n) = e^{-\beta x_n} \).

Then, from (9), we obtain the following iterative method for finding the multiple roots of the nonlinear equation.

Algorithm 2.1

For a given \( x_0 \), find the approximate solution \( x_{n+1} \) by the iterative scheme:

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

If \( \beta = 0 \), Algorithm 2.1 reduces to the well known modified Newton method (1).

If \( \beta = \frac{f''(x_n)}{2f'(x_n)} \),

then Algorithm 2.1 reduces to the well known Halley method for \( m = 1 \). (Burden and Faire, 2001; Noor, 2007; Traub, 1964). We also observe that Algorithm 2.1 is a good alternative of modified
Newton method, since this method works properly even when the value of \( f'(x_n) \) is zero at some \( x_n \).

ii) Let \( g(x_n) = e^{\beta f(x_n)} \).

Then, from (9), we have the following iterative method for finding the multiple roots of the nonlinear equation.

**Algorithm 2.2**

For a given \( x_0 \), find the approximate solution \( x_{n+1} \) by the iterative scheme:

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

If \( \beta = 0 \), then Algorithm 2.2 reduces to the well known modified Newton method (1).

If \( \beta = \frac{1}{2} f'(x_n) \), then Algorithm 2.2 reduces to the well known Halley method for \( m = 1 \), (Burden and Faire, 2001; Noor, 2007; Traub 1964).

iii) Let \( g(x_n) = e^{\beta f(x_n)} \).

Then, from (9), we have the following iterative scheme for finding the multiple roots of the nonlinear equation.

**Algorithm 2.3**

For a given \( x_0 \), find the approximate solution \( x_{n+1} \) by the iterative scheme:

\[
x_{n+1} = x_n - m \frac{f(x_n)}{f'(x_n) - \beta f''(x_n) f(x_n) + \beta \left[ \frac{f(x_n)}{f'(x_n)} \right]^2 f''(x_n)}.
\]

If \( \beta = 0 \), then Algorithm 2.3 reduces to the well known modified Newton method (1). If we neglect the last term in the denominator for Algorithm 2.3, then the Algorithm 2.3 reduces to Algorithm 2.1.

iv) Let \( g(x_n) = e^{-\beta f(x_n)} \).

Then, from (9), we have the following iterative scheme for solving the nonlinear equation of multiple roots.

**Algorithm 2.4**

For a given \( x_0 \), find the approximate solution \( x_{n+1} \) by the iterative scheme:

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

If \( \beta = 0 \), Algorithm 2.4 reduces to the well known modified Newton method (1).

If \( \beta = \frac{f''(x_n)}{2[f'(x_n)]^2} \), then Algorithm 2.2 reduces to the well known Halley method for \( m = 1 \), (Burden and Faire, 2001; Noor, 2007; Traub 1964).

For suitable and appropriate choice of the parameter \( \beta \), Haley method and its variants form can be deduced for simple roots. This novel technique was introduced and applied in Noor (2007).

**Case 2**

Now we develop an iteration relation which will generate the iterative methods of third order for obtaining multiple roots of nonlinear equations. We use the modified Newton method as \( \phi(x_n) \), the auxiliary function.

Let,

\[
\phi(x_n) = x_n - m \frac{f(x_n)}{f'(x_n)}.
\]

Then,

\[
\phi'(x_n) = 1 - m \left[ 1 - \frac{f(x_n) f''(x_n)}{[f'(x_n)]^2} \right].
\]

Using (10), (11) and \( p = 2 \) in (8), we get:

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

Equation 12 is the main recurrence relation for the iterative methods. We use (12) to suggest some iterative methods of order three for solving nonlinear equations and this is the main motivation. We now consider some special cases of the auxiliary functions \( g(x_n) \).

i) Let \( g(x_n) = e^{-\beta x_n} \).

Then, from (12), we obtain the following iterative method for finding the multiple roots of the nonlinear equations \( f(x) = 0 \).
Algorithm 2.5
For a given $x_0$, find the approximate solution $x_{n+1}$ by the iterative scheme:

$$x_{n+1} = x_n - m \frac{f(x_n)}{f'(x_n)} - \frac{m}{2} \left( (1-m) + \frac{m f(x_n) f'(x_n)}{f'(x_n)} \right) \left( \frac{f(x_n)}{f'(x_n)} - \beta f(x_n) \right).$$

ii) Let $g(x_n) = e^{f(x_n)}$.

Then, from (12), we have the following iterative method for finding the multiple roots of the nonlinear equations.

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

$$x_{n+1} = x_n - m \frac{f(x_n)}{f'(x_n)} - \frac{m}{2} \left( (1-m) + \frac{m f(x_n) f'(x_n)}{f'(x_n)} \right) \left( \frac{f(x_n)}{f'(x_n)} - \beta f(x_n) \right).$$

iii) Let $g(x_n) = e^{f(x_n)}$.

Then, from (12), we have the following iterative scheme for finding the multiple roots of the nonlinear equation $f(x) = 0$.

Algorithm 2.7
For a given $x_0$, find the approximate solution $x_{n+1}$ by the iterative scheme:

$$x_{n+1} = x_n - m \frac{f(x_n)}{f'(x_n)} - \frac{m}{2} \left( (1-m) + \frac{m f(x_n) f'(x_n)}{f'(x_n)} \right) \left( \frac{f(x_n)}{f'(x_n)} - \beta f(x_n) \right).$$

iv) Let $g(x_n) = e^{-\beta f(x_n)}$.

Then, from (12), we have the following iterative scheme for finding the multiple roots of the nonlinear equations:

$f(x) = 0$.

Algorithm 2.8
For a given $x_0$, find the approximate solution $x_{n+1}$ by the iterative scheme:

$$x_{n+1} = x_n - m \frac{f(x_n)}{f'(x_n)} - \frac{m}{2} \left( (1-m) + \frac{m f(x_n) f'(x_n)}{f'(x_n)} \right) \left( \frac{f(x_n)}{f'(x_n)} - \beta f(x_n) \right).$$

(It is needless to say that never choose such a value of $\beta$ which makes the denominator zero. It is necessary that sign of $\beta$ should be chosen so as to keep the denominator largest in magnitude in Algorithms 2.8).

CONVERGENCE ANALYSIS

Theorem 3.1
Let $\phi(x_n)$ be the arbitrary iteration function of order $p \geq 1$ at $\alpha$, where $\alpha$ is a root of $f(x)$ of $m$-fold. Suppose that $\phi(x)$ is continuously differentiable at $\alpha$, then the iteration function (8) is of order at least $p + 1$.

Proof
Here we will study the function of the type:

$$\psi(x) = \phi(x) - \frac{m \phi'(x) f(x) g(x)}{p[f'(x) g(x) + f(x) g'(x)]}. \quad (13)$$

Since $\alpha$ is the root of $f(x)$ and $\phi(x)$ is of $p$th-order convergent iteration function, so we have:

$f(\alpha) = 0, \phi(\alpha) = \alpha, \phi'(\alpha) = 0, \phi''(\alpha) = 0, ..., \phi^{(p-1)}(\alpha) = 0, \phi^{(p)}(\alpha) \neq 0$.

Let,

$$\eta(x) = \frac{f(x) g(x)}{f'(x) g(x) + f(x) g'(x)}. \quad (14)$$

Then,

$$\eta(\alpha) = 0, \eta'(\alpha) = 1, \text{ since } f(\alpha) = 0.$$

From (13) and (14), we have:

$$\psi(x) = \phi(x) - \frac{m \phi'(x) \eta(x)}{p \phi'(x) \eta(x)}. \quad (15)$$

Clearly,

$$\psi(\alpha) = \alpha, \text{ since } \eta(\alpha) = 0, \phi(\alpha) = \alpha,$$
**Table 1. Examples and comparison for \( \beta = 1 \)**

<table>
<thead>
<tr>
<th>( f(x_n) )</th>
<th>( x_0 )</th>
<th>NM</th>
<th>ECM</th>
<th>HM</th>
<th>OM</th>
<th>Alg 2.5</th>
<th>Alg 2.6</th>
<th>Alg 2.7</th>
<th>Alg 2.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>-0.3</td>
<td>52</td>
<td>8</td>
<td>57</td>
<td>26</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>43</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>2.7</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>1.8</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>( f_5 )</td>
<td>3.5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>( f_6 )</td>
<td>1.7</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>( f_7 )</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>Fail</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>( f_8 )</td>
<td>.35</td>
<td>16</td>
<td>10</td>
<td>8</td>
<td>11</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>15</td>
</tr>
<tr>
<td>( f_9 )</td>
<td>-2.5</td>
<td>9</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>( f_{10} )</td>
<td>3.2</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>( f_{11} )</td>
<td>4.2</td>
<td>20</td>
<td>14</td>
<td>11</td>
<td>15</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>19</td>
</tr>
</tbody>
</table>

Differentiating (15) with respect to \( x \), one can have:

\[
\psi^{(p)}(x) = \phi^{(p)}(x) - \frac{m}{p} \sum_{n=0}^{p} (\binom{p}{n}) \phi^{(p-n+1)}(x) \eta^{(n)}(x),
\]

(16)

And

\[
\psi^{(p+1)}(x) = \phi^{(p+1)}(x) - \frac{m}{p+1} \sum_{n=0}^{p+1} (\binom{p+1}{n}) \phi^{(p-n+2)}(x) \eta^{(n)}(x).
\]

(17)

From (16) and (17), we conclude that:

\[
\psi^{(p)}(\alpha) = 0,
\]

(18)

And

\[
\psi^{(p+1)}(\alpha) = \left[1 - m\left(1 + \frac{1}{p}\right)\right] \phi^{(p+1)}(\alpha) \neq 0,
\]

Due to the following results,

\[
\phi'(\alpha) = \phi'(\alpha) = ... \phi^{(p-1)}(\alpha) = 0, \phi^{(p+1)}(\alpha) \neq 0, \eta(\alpha) = 0 \text{ and } \eta'(\alpha) = 1.
\]

Thus, we see that \( \psi(x) \) is of order at least \( p + 1 \). □

**Special cases**

i) If \( \frac{m f(x)g(x)}{f'(x)g(x) + f(x)g'(x)} = x - \psi(x), \) in (8), then we can obtain:

\[
\psi(x) = x - p \frac{x - \phi(x)}{p - \phi'(x)},
\]

and consequently, we obtain the known result Jovanovic (1972) as a special case of Theorem 3.1.

ii) The main recurrence relation of Noor (2007) is the special case of (9) for \( m = 1 \).

iii) The main recurrence relation of Noor and Sha (2009) is the special case of (12) for \( m = 1 \).

iv) The Euler-Chebyshev third-order multiple roots finding method is the special case of Algorithms 2.5 to 2.8 for \( \beta = 0 \).

**Discussion of numerical results**

We now present some numerical test results for various third-order multiple root-finding methods and the Newton method in Tables 1 and 2 for \( \beta = 1 \) and \( \beta = 0.5 \) respectively. We compare the modified Newton method (NM), the Euler-Chebyshev method (ECM), Hally-like method (HM), Osada’s method (OM) and Algorithm 2.5, 2.6, 2.7 and 2.8 introduced in the contribution. All
Table 2. (Examples and comparison for $\beta = 0.5$)

<table>
<thead>
<tr>
<th>$f(x_n)$</th>
<th>$x_0$</th>
<th>NM</th>
<th>ECM</th>
<th>HM</th>
<th>OM</th>
<th>Alg 2.5</th>
<th>Alg 2.6</th>
<th>Alg 2.7</th>
<th>Alg 2.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>-0.3</td>
<td>52</td>
<td>8</td>
<td>57</td>
<td>26</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$f_2$</td>
<td>2.7</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>3.5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>$f_3$</td>
<td>1.8</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$f_4$</td>
<td>1.7</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>$f_5$</td>
<td>.35</td>
<td>16</td>
<td>10</td>
<td>8</td>
<td>11</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>9</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>$f_6$</td>
<td>4.2</td>
<td>20</td>
<td>14</td>
<td>11</td>
<td>15</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$f_7$</td>
<td>5.5</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

computations are done using the MAPLE using 128 digits floating point arithmetic (Digits: = 128). We will use $\varepsilon = 10^{-16}$. The following stopping criteria are used for computer programs.

(i) $|x_{n+1} - x_n| < \varepsilon$.

(ii) $|f(x_n)| < \varepsilon$.

The following functions are used for the comparison and we display the approximate zeros found, up to the 28th decimal place.

Function $m$ | Approximate solution
---|---
$f_1(x) = (x^3 + 4x^2 - 10)^3$ | $m = 3$
1.365230013414096457608068290

$f_3(x) = (x^2 - e^x - 3x + 2)^5$ | $m = 5$
0.2575302854398607604553673049

$f_4(x) = (\cos x - x)^3$ | $m = 3$
0.7390851332151606416553120876

$f_5(x) = (xe^{x^2} - \sin^2 x + 3\cos x + 5)^4$ | $m = 4$
-1.2076478271309189270094167584

$f_6(x) = (e^{x^2 + 7x - 30} - 1)^4$ | $m = 4$
3.0

$f_7(x) = (e^x + x - 2)^2$ | $m = 2$

CONCLUSION

In this paper, we have used the variational iteration method for generating a series of iterative methods for finding the multiple roots of the nonlinear equations. We have also studied the convergence analysis of the general iterative method. Several special cases are also considered. The results obtained in this may stimulate further research in this field. The interested readers are advised to discover the new and innovative applications of the variational iteration methods. Variational iteration methods have been extended and generalized for finding the approximate solution of the obstacle and unilateral problems, which can be studied in the general framework of variational problems (Noor, 1988, 2000, 2004, 2009; Noor et al., 1993, 2010).

ACKNOWLEDGEMENTS

This research was supported by the Visiting Professor Program of King Saud University, Riyadh, Saudi Arabia and Research Grant No. KSU.VPP. 108. The authors would like to thank the referees and the "editor" for their valuable comments and suggestions. The authors are
also grateful to Dr. S. M. Junaid Zaidi, Rector, CIIT for providing excellent research facilities and environment.

REFERENCES


