Research Article

Modified Homotopy Perturbation Technique for the Approximate Solution of Nonlinear Equations

Farooq Ahmed Shah

Department of Mathematics, COMSATS Institute of Information Technology, Attock Campus, Kamra Road, Attock 43600, Pakistan

Correspondence should be addressed to Farooq Ahmed Shah; farooqhamdani@gmail.com

Received 10 October 2013; Accepted 2 January 2014; Published 18 February 2014

Academic Editors: T. Calvo, Q. Guo, and X. Song

Copyright © 2014 Farooq Ahmed Shah. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We use a new modified homotopy perturbation method to suggest and analyze some new iterative methods for solving nonlinear equations. This new modification of the homotopy method is quite flexible. Various numerical examples are given to illustrate the efficiency and performance of the new methods. These new iterative methods may be viewed as an addition and generalization of the existing methods for solving nonlinear equations.

1. Introduction

It is well known that a wide class of problems, which arises in various branches of mathematical and engineering science, can be studied in the unified framework of nonlinear equation of the form $f(x) = 0$. Numerical methods for finding the approximate solutions of the nonlinear equation are being developed by using several different techniques including the Taylor series, quadrature formulas, homotopy, and decomposition techniques; see [1–14] and the references therein.

In this paper, we use the modified homotopy perturbation technique to suggest the main iterative schemes which generates the iterative methods of higher order. First of all, we rewrite the given nonlinear equation along with the auxiliary function $g(x) = 0$, as an equivalent coupled system of equations using the Taylor series. This approach enables us to express the given nonlinear equation as sum of linear and nonlinear equations. This way of writing the given equation is known as the decomposition and plays the central role in suggesting the iterative methods for solving nonlinear equations $f(x) = 0$. Results obtained in this paper suggest that this new technique of decomposition is a promising tool. In Section 2, we outline the main ideas of this technique and suggest one-step and two-step iterative methods for solving nonlinear equations. One can notice that if the derivative of the function vanishes, that is, $|f'(x)| = 0$, during the

iterative process, then the sequence generated by the Newton method or the methods derived in [1–9] are not defined. Due to such cardinal sin of division results a mathematical breakdown. This is another motivation of the paper that the derived higher order methods also converge even if the derivative vanishes during the iterative process. Several numerical examples are given to illustrate the efficiency and the performance of the new iterative methods. Our results can be considered as an important improvement and refinement of the previously known results.

2. Construction of Iterative Methods

In this section, we construct some higher ordered iterative methods for solving nonlinear equations. For this purpose, we use the idea of modified homotopy perturbation method.

Consider the nonlinear equation of the type

$$f(x) = 0. \quad (1)$$

Assume that r is a simple root of nonlinear equation (1) and γ is an initial guess sufficiently close to r . Let $g(x)$ be the auxiliary function, such that

$$f(x)g(x) = 0. \quad (2)$$

We can rewrite the nonlinear equation (2) as a system of coupled equations using the Taylor series technique as

$$f(\gamma)g(\gamma) + [f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)](x - \gamma) + h(x) = 0 \quad (3)$$

or

$$h(x) = f(x)g(\gamma) - f(\gamma)g(\gamma) - [f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)](x - \gamma), \quad (4)$$

where γ is the initial approximation for a zero of (1). We can rewrite (4) in the following form:

$$x = \gamma - \frac{f(\gamma)g(\gamma)}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]} - \frac{h(x)}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]}. \quad (5)$$

From the relation (4), it is clear that

$$h(x_0) = f(x_0)g(\gamma). \quad (6)$$

It is important to note that (6) plays an important role in the derivation of the iterative methods; see Chun [3] and Noor [11]. We rewrite (5) in the following form:

$$x = c + N(x), \quad (7)$$

where

$$c = \gamma - \frac{f(\gamma)g(\gamma)}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]}, \quad (8)$$

$$N(x) = -\frac{h(x)}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]}. \quad (9)$$

Here $N(x)$ is a nonlinear operator. We use modified homotopy technique to develop iterative schemes for solving nonlinear equations.

Define

$$H(v, p, T) : (R \times [0, 1]) \times R \longrightarrow R \quad (10)$$

as

$$H(v, p, T) = (1 - p)[L(v) - L(x_0)] + p[L(v) - c - N(v)] - p(1 - p)T = 0 \quad (11)$$

or

$$H(v, p, T) = L(v) - L(x_0) + pL(x_0) + p[-c + N(v)] - p(1 - p)T = 0, \quad (12)$$

where $p \in [0, 1]$ is called embedding parameter, x_0 is an initial approximation, and T is an arbitrary auxiliary operator.

We would like to emphasize that one has opportunity to select the initial guess and auxiliary operator T . From (12), we get

$$H(v, 0, T) = L(v) - L(x_0) = v - c = 0, \quad (13)$$

$$H(v, 1, T) = L(v) - c - N(v) = v - c - N(v) = 0.$$

The embedding parameter p monotonically increases from zero to unity as the trivial problem $H(v, 0, T) = L(v) - L(x_0) = v - c = 0$ is continuously deformed to the original problem $H(v, 1, T) = L(v) - c - N(v) = v - c - N(v) = 0$. The changing process of p from zero to unity is called deformation and $L(v) - L(x_0)$ and $L(v) - c - N(v)$ are homotopic.

For the application of modified homotopy perturbation method to (1), we can write (12) as follows by expanding $N(x_0)$ into a Taylor series at x_0 :

$$v - c = p \left[N(x_0) + (v - x_0)N'(x_0) + (v - x_0)^2 \frac{N''(x_0)}{2} + L \right] + p(1 - p)T. \quad (14)$$

Now substituting (8) into (12), we have

$$x_0 + x_1p + x_2p^2 + L - c = p \left[N(x_0) + (x_1p + x_2p^2 + L)N'(x_0) + (x_1p + x_2p^2 + L)^2 \frac{N''(x_0)}{2} + L \right] + p(1 - p)T. \quad (15)$$

By equating the coefficients of the same powers of p , we have

$$p^0 : x_0 - c, \quad (16)$$

$$p^1 : x_1 = N(x_0) + T, \quad (17)$$

$$p^2 : x_2 = x_1N'(x_0) - T, \quad (18)$$

$$p^3 : x_3 = x_2N'(x_0) + x_1^2 \frac{N''(x_0)}{2}. \quad (19)$$

To find auxiliary parameter T , we take $x_2 = 0$. In this case, from (18), we have

$$x_1 = N(x_0) + T. \quad (20)$$

By substituting (19) into (18), we have

$$x_2 = [(N'(x_0) + T)N'(x_0)] - T. \quad (21)$$

Setting $x_2 = 0$, we obtain

$$T = \frac{N(x_0)N'(x_0)}{1 - N'(x_0)}. \quad (22)$$

Substituting (22) into (18), we have

$$x_1 = \frac{N(x_0)}{1 - N'(x_0)}. \quad (23)$$

Again substituting $x_2 = 0$, in (19), we get

$$x_3 = \left[\frac{N(x_0)}{1 - N'(x_0)} \right]^2 \frac{N''(x_0)}{2}. \quad (24)$$

From (9), we have

$$N(x_0) = -\frac{h(x_0)}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]}. \quad (25)$$

One can easily compute

$$\begin{aligned} N'(x_0) &= 1 - \frac{f'(x_0)g(\gamma)}{[f'(x_0)g(\gamma) + f(\gamma)g'(\gamma)]}, \\ N''(x_0) &= -\frac{f''(x_0)g(\gamma)}{[f''(x_0)g(\gamma) + f(\gamma)g'(\gamma)]}. \end{aligned} \quad (26)$$

Note that x is approximated by

$$x = \lim_{m \rightarrow \infty} X_m = x_0 + x_1 + x_2 + \cdots + x_m. \quad (27)$$

For $m = 0$, we have

$$x \approx x_0 = \gamma - \frac{f(\gamma)g(\gamma)}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]}, \quad (28)$$

which gives the following iterative method for solving nonlinear equations.

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

$$\begin{aligned} x_{n+1} &= x_n - \frac{f(x_n)g(x_n)}{[f'(x_n)g(x_n) + f(x_n)g'(x_n)]}, \\ n &= 0, 1, 2, 3, \dots \end{aligned} \quad (29)$$

This is the main iteration scheme which generates at least quadratically convergent iterative methods for solving nonlinear equations. This scheme is also introduced by He [9] and Noor [10] for this purpose by using different technique.

For $m = 1$, we have

$$\begin{aligned} x &\approx x_0 + x_1 = x_0 + N(x_0), \\ x &\approx x_0 = \gamma - \frac{f(\gamma)g(\gamma)}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]} \\ &\quad - \frac{f(x_0)g(\gamma)}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]}. \end{aligned} \quad (30)$$

This formulation allows us the following iterative scheme for solving nonlinear equations.

Algorithm 2. For a given x_0 , find the approximation solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{[f'(x_n)g(x_n) + f(x_n)g'(x_n)]}, \\ x_{n+1} &= y_n - \frac{f(y_n)g(x_n)}{[f'(x_n)g(x_n) + f(x_n)g'(x_n)]}. \end{aligned} \quad (31)$$

In a similar way, for $m = 3$, we have

$$\begin{aligned} x &\approx x_0 + x_1 + x_3 = x_0 + N'(x_0) + \frac{x_1^2}{2}N''(x_0) \\ &= \gamma - \frac{f(\gamma)g(\gamma)}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]} \\ &\quad - \frac{f(x_0)g(\gamma)}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]} \\ &\quad - \frac{f(x_0)g(\gamma)}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]} \\ &\quad + \frac{f(x_0)f'(x_0)g(\gamma)^2}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]^2} \\ &\quad - \frac{1}{2} \frac{(f(x_0)g(\gamma))^2}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]^2} \\ &\quad \times \frac{f''(x_0)g(\gamma)}{[f'(\gamma)g(\gamma) + f(\gamma)g'(\gamma)]}. \end{aligned} \quad (32)$$

This formulation allows us the following iterative method for solving nonlinear equations.

Algorithm 3. For a given x_0 , find the approximation solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{[f'(x_n)g(x_n) + f(x_n)g'(x_n)]}, \\ x_{n+1} &= y_n - \frac{2f(y_n)g(x_n)}{[f'(x_n)g(x_n) + f(x_n)g'(x_n)]} \\ &\quad + \frac{f(y_n)f'(y_n)g(x_n)^2}{[f'(x_n)g(x_n) + f(x_n)g'(x_n)]^2} \\ &\quad - \frac{1}{2} \frac{(f(y_n)g(x_n))^2}{[f'(x_n)g(x_n) + f(x_n)g'(x_n)]^2} \\ &\quad \times \frac{f''(y_n)g(x_n)}{[f'(x_n)g(x_n) + f(x_n)g'(x_n)]}. \end{aligned} \quad (33)$$

For $f''(y_n) = 0$, Algorithm 3 reduces to the following new iterative method for solving nonlinear equations.

Algorithm 4. For a given x_0 , find the approximation solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)g(x_n)}{[f'(x_n)g(x_n) + f(x_n)g'(x_n)]}, \\ x_{n+1} &= y_n - \frac{2f(y_n)g(x_n)}{[f'(x_n)g(x_n) + f(x_n)g'(x_n)]} \\ &\quad + \frac{f(y_n)f'(y_n)g(x_n)^2}{[f'(x_n)g(x_n) - f(x_n)g'(x_n)]^2}. \end{aligned} \quad (34)$$

In all the above derived iterative methods, there is an arbitrary function $g(x_n)$ involved. This arbitrary function helps to produce the methods for best implementation. If we take $g(x_n)$ as a constant function, then we obtain the following methods from Algorithms 5–8.

Algorithm 5. For a given x_0 , find the approximation solution x_{n+1} by the following iterative scheme:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, 3, \dots \quad (35)$$

Algorithm 6. For a given x_0 , find the approximation solution x_{n+1} by the following iterative schemes:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \quad x_{n+1} = y_n - \frac{f(y_n)}{f'(x_n)}. \quad (36)$$

Algorithm 7. For a given x_0 , find the approximation solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n - \frac{2f(y_n)}{f'(x_n)} + \frac{f(y_n)f'(y_n)}{[f'(x_n)]^2} \\ &\quad - \frac{1}{2} \frac{(f(y_n))^2}{[f'(x_n)]^2} \frac{f''(y_n)}{[f'(x_n)]}. \end{aligned} \quad (37)$$

Algorithm 8. For a given x_0 , find the approximation solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n - \frac{2f(y_n)}{f'(x_n)} + \frac{f(y_n)f'(y_n)}{[f'(x_n)]^2}. \end{aligned} \quad (38)$$

One can select different value of the auxiliary function $g(x_n)$ and obtain various iterative schemes for solving nonlinear equations. For the sake of completeness and to convey the implementation idea we take $g(x_n) = e^{-tx_n}$ and obtain the following iterative methods from Algorithms 1–4.

Algorithm 9. For a given x_0 , find the approximation solution x_{n+1} by the following iterative scheme:

$$x_{n+1} = x_n - \frac{f(x_n)}{[f'(x_n) - tf(x_n)]}, \quad n = 0, 1, 2, 3, \dots \quad (39)$$

This is the main iteration scheme which is also introduced by He [9] and Noor [10] for generating different iterative methods for solving nonlinear equations by using different technique.

Algorithm 10. For a given x_0 , find the approximation solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{[f'(x_n) - tf(x_n)]}, \\ x_{n+1} &= y_n - \frac{f(y_n)}{[f'(x_n) - tf(x_n)]}. \end{aligned} \quad (40)$$

Algorithm 11. For a given x_0 , find the approximation solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{[f'(x_n) - tf(x_n)]}, \\ x_{n+1} &= y_n - \frac{2f(y_n)}{[f'(x_n) - tf(x_n)]} + \frac{f(y_n)f'(y_n)}{[f'(x_n) - tf(x_n)]^2} \\ &\quad - \frac{1}{2} \frac{(f(y_n))^2}{[f'(x_n) - tf(x_n)]^2} \frac{f''(y_n)}{[f'(x_n) - tf(x_n)]}. \end{aligned} \quad (41)$$

Algorithm 12. For a given x_0 , find the approximation solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{[f'(x_n) - tf(x_n)]}, \\ x_{n+1} &= y_n - \frac{2f(y_n)}{[f'(x_n) - tf(x_n)]} + \frac{f(y_n)f'(y_n)}{[f'(x_n) - tf(x_n)]^2}. \end{aligned} \quad (42)$$

Remark 13. We would like to point out that, if we take $t = 0$, $t = 1/2$, $t = 1, \dots$, in Algorithms 1–4, we can obtain various classes of known and new iterative methods for solving nonlinear equations.

Remark 14. It is important to say to never choose such a value of $t \in \mathbb{R}$ which makes the denominator zero. It is necessary that sign of t should be chosen so as to keep the denominator largest in magnitude in the above derived Algorithms.

3. Convergence Analysis

In this section, we consider the convergence criteria of the main iterative schemes Algorithms 5 and 6 developed in Section 2.

Theorem 15. Assume that the function $f : \mathcal{D} \subset \mathbb{R} \rightarrow \mathbb{R}$ for an open interval in $\mathcal{D}$ with simple root $r \in \mathcal{D}$. Let $f(x)$ be a smooth sufficiently in some neighborhood of root and then by Algorithm 4 one sees that it has fourth-order convergence.

Proof. Let r be a simple root of the nonlinear equation $f(x)$. Since f is sufficiently differentiable, expanding $f(x_n)$ and $f'(x_n)$ in Taylor's series at r , we obtain

$$\begin{aligned} f(x_n) &= f'(r) \left[e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 \right. \\ &\quad \left. + c_5 e_n^5 + c_6 e_n^6 + O(e_n^7) \right], \\ f'(x_n) &= f'(r) \left[1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 \right. \\ &\quad \left. + 5c_5 e_n^4 + 6c_6 e_n^5 + O(e_n^7) \right], \end{aligned} \quad (43)$$

where

$$e_n = x_n - r, \quad c_k = \frac{f^{(k)}(r)}{k! f'(r)}, \quad k = 2, 3, \dots \quad (44)$$

Now expanding $f(x_n)g(x_n)$, $f(x_n)g'(x_n)$, and $f'(x_n)g'(x_n)$ in Taylor's series about r , we obtain

$$\begin{aligned} f(x_n)g(x_n) &= f(r) \left[g(r)e_n + (g(r) + c_2 g'(r))e_n^2 \right. \\ &\quad \left. + g_1 e_n^3 + O(e_n^4) \right], \end{aligned} \quad (45)$$

$$\begin{aligned} f(x_n)g'(x_n) &= f(r) \left[g'(r)e_n + (g'(r) + c_2 g''(r))e_n^2 \right. \\ &\quad \left. + g'_1 e_n^3 + O(e_n^4) \right], \end{aligned} \quad (46)$$

$$\begin{aligned} f'(x_n)g'(x_n) &= f(r) \left[(g(r) + g'(r) + 2c_2 g(r))e_n \right. \\ &\quad \left. + (g_1)e_n^2 + g_2 e_n^3 + O(e_n^4) \right], \end{aligned} \quad (47)$$

where

$$\begin{aligned} g_1 &= \frac{1}{2} g''(r) + c_3 g'(r) + c_2 g(r), \\ g_2 &= \frac{1}{6} g'''(r) + 4c_4 g'(r) + c_2 g''(r) + 3c_3 g'(r). \end{aligned} \quad (48)$$

From (45), (46), and (47), we get

$$\begin{aligned} &\frac{f(x_n)g(x_n)}{[f'(x_n)g(x_n) + f(x_n)g'(x_n)]} \\ &= e_n - \left(\frac{g'(r)}{g(r)} + c_2 \right) e_n^2 \\ &\quad - \left(\frac{g''(r)}{g(r)} + c_2^2 + 2c_3 - 2 \frac{g'(r)}{g(r)} c_2 \right. \\ &\quad \left. + 2c_2 + \frac{g'(r)}{g(r)} \right) e_n^3 + O(e_n^4). \end{aligned} \quad (49)$$

Using (49), we have

$$\begin{aligned} y_n &= r + \left(\frac{g'(r)}{g(r)} + c_2 \right) e_n^2 \\ &\quad - \left(\frac{g''(r)}{g(r)} + c_2^2 + 2c_3 - 2 \frac{g'(r)}{g(r)} c_2 + 2c_2 + \frac{g'(r)}{g(r)} \right) e_n^3 \\ &\quad + O(e_n^4). \end{aligned} \quad (50)$$

Expanding $f(y_n)$ in Taylor's series about r and using (50), we have

$$\begin{aligned} f(y_n) &= f'(r) \left[\left(\frac{g'(r)}{g(r)} + c_2 \right) e_n^2 \right. \\ &\quad \left. - \left\{ \frac{g''(r)}{g(r)} + c_2^2 + 2c_3 - 2 \frac{g'(r)}{g(r)} c_2 \right. \right. \\ &\quad \left. \left. + 2c_2 + \frac{g'(r)}{g(r)} \right\} e_n^3 + O(e_n^4) \right]. \end{aligned} \quad (51)$$

Expanding $f'(y_n)$ in Taylor's series about r and using (50), we have

$$\begin{aligned} f'(y_n) &= f'(r) \left[1 + 2c_2 \left(\frac{g'(r)}{g(r)} + c_2 \right) e_n^2 - 2c_2 \right. \\ &\quad \times \left\{ \frac{g''(r)}{g(r)} + c_2^2 + 2c_3 - 2 \frac{g'(r)}{g(r)} c_2 \right. \\ &\quad \left. \left. + 2c_2 + \frac{g'(r)}{g(r)} \right\} e_n^3 + O(e_n^4) \right]. \end{aligned} \quad (52)$$

Using (51), we get

$$\begin{aligned} &f(y_n)g(x_n) \\ &= f'(r) \left[(g'(r) + c_2 g(r)) e_n^2 \right. \\ &\quad \left. - \left\{ \frac{g'(r)^2}{g(r)} + c_2 g(r) - g''(r) + 2c_2 g(r) \right. \right. \\ &\quad \left. \left. - 2c_3 g(r) \frac{g'(r)^2}{g(r)} \right\} e_n^3 + O(e_n^4) \right]. \end{aligned} \quad (53)$$

Now using (51) and (52), we get

$$\begin{aligned} &f(y_n)f'(y_n)[g(x_n)]^2 \\ &= [f'(r)]^2 \left[(g'(r) + c_2 g(r)) g(r) e_n^2 - g(r) \right. \\ &\quad \times \left\{ g''(r) - 2c_2^2 g(r) + 2c_3 g(r) \right\} e_n^3 \\ &\quad \left. + O(e_n^4) \right]. \end{aligned} \quad (54)$$

Now using (45), (46), and (51), we get

$$\begin{aligned} & \frac{f(y_n)g(x_n)}{[f'(x_n)g(x_n) + f(x_n)g'(x_n)]} \\ &= f(r) \left[\left(\frac{g'(r)}{g(r)} + c_2 \right) e_n^2 \right. \\ & \quad \left. + \left(\frac{g''(r)}{g(r)} - 4c_2^2 + 2c_3 - 5\frac{g'(r)}{g(r)}c_2 \right. \right. \\ & \quad \left. \left. - 3\frac{g'(r)}{g(r)} \right) e_n^3 \right] + O(e_n^4). \end{aligned} \quad (55)$$

Using (45), (46), (49), (54), and (55), we obtain

$$\begin{aligned} x_{n+1} = r + & \left[\left(\frac{g'(r)}{g(r)} \right)^3 + 6 \left(\frac{g'(r)}{g(r)} \right)^2 c_2 \right. \\ & \left. + 10c_2^2 \left(\frac{g'(r)}{g(r)} \right) + 5c_2^3 \right] e_n^4 \\ & + O(e_n^5). \end{aligned} \quad (56)$$

Finally, the error equation is

$$\begin{aligned} e_{n+1} = & \left[\left\{ \frac{g'(r)}{g(r)} + c_2 \right\} \left\{ 5c_2^2 + 5c_2 \frac{g'(r)}{g(r)} + \left(\frac{g'(r)}{g(r)} \right)^2 \right\} \right] e_n^4 \\ & + O(e_n^5). \end{aligned} \quad (57)$$

Error equation (57) shows that the main iterative scheme established in Section 2 as Algorithm 4 has at least fourth-order convergence and, consequently, the methods derived from this scheme will have at least fourth-order convergence. $\square$

In a similar way, one can prove the convergence of all newly derived iterative methods.

Remark 16. From the study of convergence analysis, we note that for different values of the auxiliary function $g(x_n)$, various iterative methods of higher order can be derived from Algorithms 1, 2, 3, and 4 for solving nonlinear equations.

If we select $g'(x_n)/g(x_n) = -f''(x_n)/2f'(x_n)$, then Algorithms 1, 2, and 4 reduce to the following higher order iterative methods.

Algorithm 17. For a given x_0 , find the approximation solution x_{n+1} by the following iterative scheme:

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2[f'(x_n)]^2 - f''(x_n)f(x_n)}, \quad (58)$$

which is the well known Halley method [14] of third-order convergence.

Algorithm 18. For a given x_0 , find the approximation solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{2f(x_n)f'(x_n)}{2[f'(x_n)]^2 - f''(x_n)f(x_n)}, \\ x_{n+1} &= y_n - \frac{2f(y_n)f'(y_n)}{2[f'(y_n)]^2 - f''(y_n)f(y_n)}, \end{aligned} \quad (59)$$

which is the fourth-order convergent method for solving nonlinear equations and appears to be a new one.

Algorithm 19. For a given x_0 , find the approximation solution x_{n+1} by the following iterative schemes:

$$\begin{aligned} y_n &= x_n - \frac{2f(x_n)f'(x_n)}{2[f'(x_n)]^2 - f''(x_n)f(x_n)}, \\ x_{n+1} &= y_n - \frac{4f(y_n)f'(y_n)}{2[f'(y_n)]^2 - f''(y_n)f(y_n)} \\ & \quad - \frac{4f(y_n)f'(y_n)[f'(x_n)]^2}{(2[f'(x_n)]^2 - f''(x_n)f(x_n))^2}, \end{aligned} \quad (60)$$

which is the fifth-order convergent method for solving nonlinear equations and appears to be a new one.

4. Numerical Results

We now present some examples to illustrate the efficiency of the newly developed two-step iterative methods in this paper. We compare the Newton method [14] (NM), the method of Hasanov et al. [7] (HM), the method of Chun [3] (CM), Algorithm 5 (SH1), and Algorithm 6 (SH2) introduced in this paper. We use $\epsilon = 10^{-15}$.

The following stopping criteria are used for computer programs:

$$(i) |x_{n+1} - x_n| \leq \epsilon, \quad (ii) |f(x_n)| \leq \epsilon. \quad (61)$$

We consider the following nonlinear equations as test problems.

Example 20. We consider the nonlinear equation

$$f_1(x) = \sin^2 x - x^2 + 1. \quad (62)$$

We consider the different values of parameter t , for all the methods to compare the numerical results in Tables 1 and 2, respectively.

Table 1 depicts the numerical results of Example 20. We use the initial guess $x_0 = -1$ for the computer program for $t = 1$.

Table 2 depicts the numerical results of Example 20. We use the initial guess $x_0 = -1$, for the computer program for $t = 1/2$.

TABLE 1

Method	IT	x_n (approximate solution)	$ f(x_n) $	δ	ρ
NM	7	1.4044916482153412260	$1.04e - 50$	$7.33e - 26$	2.00003
HM	5	1.4044916482153412260	0	$6.98e - 32$	3.03470
CM	5	1.4044916482153412260	0	$1.31e - 17$	3.88844
SH1	4	1.4044916482153412260	0	$7.31e - 32$	3.03479
SH2	4	1.4044916482153412260	0	$2.95e - 25$	4.13335

TABLE 2

Method	IT	x_n (approximate solution)	$ f(x_n) $	δ	ρ
NM	7	1.4044916482153412260	$1.04e - 50$	$7.33e - 26$	2.00003
HM	5	1.4044916482153412260	0	$6.98e - 32$	3.03470
CM	5	1.4044916482153412260	0	$1.31e - 17$	3.88844
SH1	5	1.4044916482153412260	0	$7.31e - 32$	3.03271
SH2	4	1.4044916482153412260	0	$2.95e - 25$	4.17535

TABLE 3

Method	IT	x_n (approximate solution)	$ f(x_n) $	δ	ρ
NM	6	0.25753028543986076045	$2.93e - 55$	$9.10e - 28$	2.00050
HM	4	0.25753028543986076045	$3.9e - 53$	$1.06e - 17$	3.49362
CM	4	0.25753028543986076045	0	$9.46e - 29$	4.37143
SH1	3	0.25753028543986076045	$3.38e - 5$	$1.01e - 17$	3.28939
SH2	4	0.25753028543986076045	0	$1.08e - 45$	4.34860

Example 21. We consider the nonlinear equation

$$f_2(x) = x^2 - e^{-x} - 3x + 2. \quad (63)$$

We consider $t = 1$ and $t = 1/2$, for all the methods to compare the numerical results in Tables 3 and 4, respectively.

Table 3 shows the efficiency of the methods for Example 21. We use the initial guess $x_0 = 2$, for the computer program for $t = 1$. The number of iterations and computational order of convergence give us an idea about the better performance of the new methods.

Table 4 shows the efficiency of the methods for Example 21. We use the initial guess $x_0 = 2$, for the computer program for $t = 0.5$.

Example 22. We consider the nonlinear equation

$$f_3(x) = (x - 1)^3 - 1. \quad (64)$$

We consider $t = 1$ and $t = 0.5$, for all the methods to compare the numerical results in Tables 5 and 6, respectively.

In Table 5, the numerical results for Example 22 are described. We use the initial guess $x_0 = 1.5$ for the computer program for $t = 0$. We observe that all the methods approach the approximate solution after an equal number of iterations, but the computational order of convergence has little bit difference.

In Table 6, the numerical results for Example 22 are described. We use the initial guess $x_0 = 1.5$ for the computer program.

Example 23. We consider the nonlinear equation

$$f_4(x) = xe^{x^2} - \sin^2 x + 3 \cos x + 5. \quad (65)$$

We consider $t = 1$ and $t = 0.5$, for all the methods to compare the numerical results in Tables 7 and 8, respectively.

Table 7 shows the numerical results for Example 23. For the computer program we use the initial guess $x_0 = -2$ and $t = 1$. We note that the new derived methods have better computational order of convergence and approach the desired result in a less number of iterations.

Table 8 shows the numerical results for Example 23. For the computer program we use the initial guess $x_0 = -2$, and $t = 1$. We note that the new derived methods have better computational order of convergence and approach to the desired result in less number of iterations.

Example 24. We consider the nonlinear equation

$$f_5(x) = e^{x^2+7x-30} - 1. \quad (66)$$

We consider $t = 1$ and $t = 0.5$, for all the methods to compare the numerical results in Tables 9 and 10, respectively.

In Table 9, we show the numerical results for Example 24. We use the initial guess $x_0 = 5$ and $t = 1$, for the computer program. We observe that the new methods approach the desired approximate solution in an 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.

In Table 10, we show the numerical results for the Example 24. We use the initial guess $x_0 = 5$ and $t = 0.5$, for

TABLE 4

Method	IT	x_n (approximate solution)	$ f(x_n) $	δ	ρ
NM	6	0.25753028543986076045	$2.93e - 55$	$9.10e - 28$	2.00050
HM	4	0.25753028543986076045	$3.9e - 53$	$1.06e - 17$	3.49362
CM	4	0.25753028543986076045	0	$9.46e - 29$	4.37143
SH1	5	0.25753028543986076045	$3.38e - 53$	$1.01e - 17$	3.28939
SH2	4	0.25753028543986076045	0	$1.08e - 45$	4.14966

TABLE 5

Method	IT	x_n (approximate solution)	$ f(x_n) $	δ	ρ
NM	8	2.00000000000000000000	$2.06e - 42$	$8.28e - 22$	2.00025
HM	5	2.00000000000000000000	0	$1.10e - 40$	2.99061
CM	5	2.00000000000000000000	0	$2.74e - 24$	3.53144
SH1	5	2.00000000000000000000	0	$1.10e - 40$	3.09063
SH2	5	2.00000000000000000000	0	$9.78e - 43$	3.86663

TABLE 6

Method	IT	x_n (approximate solution)	$ f(x_n) $	δ	ρ
NM	8	2.00000000000000000000	$2.06e - 42$	$8.28e - 22$	2.00025
HM	6	2.00000000000000000000	0	$1.10e - 40$	2.99061
CM	5	2.00000000000000000000	0	$2.74e - 24$	3.53144
SH1	5	2.00000000000000000000	0	$1.10e - 40$	2.99069
SH2	4	2.00000000000000000000	0	$9.78e - 43$	3.89667

TABLE 7

Method	IT	x_n (approximate solution)	$ f(x_n) $	δ	ρ
NM	9	-1.207647827130918927	$2.27e - 40$	$2.73e - 21$	2.00085
HM	6	-1.207647827130918927	$2.38e - 57$	$3.68e - 20$	3.00846
CM	6	-1.207647827130918927	$1.10e - 58$	$2.15e - 36$	3.88967
SH1	6	-1.207647827130918927	$2.14e - 57$	$3.56e - 20$	3.03831
SH2	5	-1.207647827130918927	$8.00e - 59$	$4.39e - 21$	4.01292

TABLE 8

Method	IT	x_n (approximate solution)	$ f(x_n) $	δ	ρ
NM	9	-1.207647827130918927	$2.27e - 40$	$2.73e - 21$	2.00085
HM	6	-1.207647827130918927	$2.38e - 57$	$3.68e - 20$	3.00846
CM	6	-1.207647827130918927	$1.10e - 58$	$2.15e - 36$	3.88967
SH1	6	-1.207647827130918927	$2.14e - 57$	$3.56e - 20$	3.00832
SH2	5	-1.207647827130918927	$8.00e - 56$	$4.39e - 22$	4.01293

TABLE 9

Method	IT	x_n (approximate solution)	$ f(x_n) $	δ	ρ
NM	13	3.00000000000000000000	$1.52e - 47$	$4.21e - 25$	2.00023
HM	9	3.00000000000000000000	0	$6.57e - 39$	2.99432
CM	8	3.00000000000000000000	0	$2.12e - 23$	3.88024
SH1	9	3.00000000000000000000	0	$6.03e - 39$	2.99434
SH2	8	3.00000000000000000000	0	$1.31e - 24$	3.93853

TABLE 10

Method	IT	x_n (approximate solution)	$ f(x_n) $	δ	ρ
NM	13	3.00000000000000000000	$1.52e - 47$	$4.21e - 25$	2.00023
HM	9	3.00000000000000000000	0	$6.57e - 39$	2.99432
CM	8	3.00000000000000000000	0	$2.12e - 23$	3.88024
SH1	9	3.00000000000000000000	0	$6.13e - 39$	3.09401
SH2	8	3.00000000000000000000	0	$1.35e - 24$	3.96253

the computer program. We calculate the computational order of convergence for all the methods which verify the rate of convergence and efficiency of the methods.

In Tables 1–10, x_n is the approximate solution at n th iteration and δ is the difference of the last two consecutive iterations. The computational order of convergence ρ is approximated for all the examples in all the tables by means of

$$\rho = \frac{\ln(|x_{n+1} - x_n| / |x_n - x_{n-1}|)}{\ln(|x_n - x_{n-1}| / |x_{n-1} - x_{n-2}|)}. \quad (67)$$

5. Conclusion

In this paper, we have studied some new iterative methods for solving nonlinear equations by using modified homotopy perturbation technique. Our method of derivation of the iterative methods is very simple as compared to the other techniques, specially the Adomian decomposition technique. If we consider the definition of efficiency index [14] as $p^{1/m}$, where p is the order of the method and m is the number of functional evaluations per iteration required by the method, we have that Algorithm 2 has the efficiency index equal to $3^{1/3} \approx 1.44224957$ while Algorithm 3 has the efficiency index $4^{1/4} \approx 1.4113562$, which is the same as that of the Newton method. The methods derived above are well defined and robust even when $f'(x_n) \approx 0$, and numerical stability of the methods can also be observed during numerical experimentations. Using the technique and idea of this paper, one can suggest and analyze higher order multistep iterative methods for solving nonlinear equations as well as the system of nonlinear equations.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

- [1] S. Abbasbandy, "Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method," *Applied Mathematics and Computation*, vol. 145, no. 2-3, pp. 887–893, 2003.
- [2] S. Abbasbandy, Y. Tan, and S. J. Liao, "Newton-homotopy analysis method for nonlinear equations," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1794–1800, 2007.
- [3] C. Chun, "A method for obtaining iterative formulas of order three," *Applied Mathematics Letters*, vol. 20, no. 11, pp. 1103–1109, 2007.
- [4] C. Chun, "Iterative methods improving newton's method by the decomposition method," *Computers and Mathematics with Applications*, vol. 50, no. 10–12, pp. 1559–1568, 2005.
- [5] W. Gautschi, *Numerical Analysis: An Introduction*, Birkhäuser, Boston, Mass, USA, 1997.
- [6] M. Grau and M. Noguera, "A variant of Cauchy's method with accelerated fifth-order convergence," *Applied Mathematics Letters*, vol. 17, no. 6, pp. 509–517, 2004.
- [7] V. I. Hasanov, I. G. Ivanov, and G. Nedjibov, "A new modification of Newton's method," in *Applications of Mathematics in Engineering and Economics*, vol. 27, pp. 278–286, 2002.
- [8] J.-H. He, "Homotopy perturbation technique," *Computer Methods in Applied Mechanics and Engineering*, vol. 178, no. 3-4, pp. 257–262, 1999.
- [9] J.-H. He, "Variational iteration method—some recent results and new interpretations," *Journal of Computational and Applied Mathematics*, vol. 207, no. 1, pp. 3–17, 2007.
- [10] M. A. Noor, "New classes of iterative methods for nonlinear equations," *Applied Mathematics and Computation*, vol. 191, no. 1, pp. 128–131, 2007.
- [11] M. A. Noor, "Iterative methods for nonlinear equations using homotopy perturbation technique," *Applied Mathematics & Information Sciences*, vol. 4, no. 2, pp. 227–235, 2010.
- [12] M. A. Noor and F. A. Shah, "Variational iteration technique for solving nonlinear equations," *Journal of Applied Mathematics and Computing*, vol. 31, no. 1-2, pp. 247–254, 2009.
- [13] M. A. Noor, F. A. Shah, K. I. Noor, and E. Al-Said, "Variational iteration technique for finding multiple roots of nonlinear equations," *Scientific Research and Essays*, vol. 6, no. 6, pp. 1344–1350, 2011.
- [14] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, New York, NY, USA, 1964.