

RESEARCH NOTE

Open Access



On new sixth and seventh order iterative methods for solving non-linear equations using homotopy perturbation technique

Srinivasarao Thota^{1*}  and P. Shanmugasundaram²

Abstract

Objectives: This paper proposes three iterative methods of order three, six and seven respectively for solving non-linear equations using the modified homotopy perturbation technique coupled with system of equations. This paper also discusses the analysis of convergence of the proposed iterative methods.

Results: Several numerical examples are presented to illustrate and validation of the proposed methods. Implementation of the proposed methods in Maple is discussed with sample computations.

Keywords: Iterative methods, Nonlinear equations, Order of convergence, Homotopy perturbation technique

Mathematics subject classification: 65N30, 49M37

Introduction

The applications of non-linear equations of the type $f(x) = 0$ arise in various branches of pure and applied sciences, engineering and computing. In recent time, several scientists and engineers have been focused to solve non-linear equations numerically as well as analytically. In the literature, there are several iterative methods/algorithms available which are derived from different techniques such as homotopy, interpolation, Taylor's series, quadrature formulas, decomposition etc., and also available various modifications and improvements of the existing methods, and different hybrid iterative methods, see, for example [1, 4–7, 9–16, 28–32, 36–38]. In general, the roots of non-linear or transcendental equations cannot be expressed in closed form or cannot be computed analytically. The root-finding algorithms provide us to compute approximations to the roots, these approximations

are expressed either as small isolating intervals or as floating point numbers. In this paper, we use the modified homotopy perturbation technique (HPT) to create a number of iterative methods for solving the given non-linear equations with converging order more than or equal to three. The given non-linear equations are expressed as an equivalent coupled system of equations with help of the Taylor's series and technique of He [4]. This enables us to express the given non-linear equation as a sum of linear and non-linear equations. The Maple implementation of the proposed algorithm is also discussed, and various Maple implementations for differential and transcendental equations are available in the literature, see, for example [17–27].

The rest of paper is organized as follows: Section recalls the preliminary concepts related to the topic; In Section , we present the methodology and steps involving in the proposed algorithms; Section discusses the analysis of convergence to show the order of proposed methods are more than or equal to three; Section presents several numerical examples to illustrate and validate the proposed methods/algorithms; and finally Section presents

*Correspondence: srinithota@gmail.com; srinivasarao.thota@sru.edu.in

¹ Department of Mathematics, School of Sciences, SR University, Warangal, Telangana 506371, India

Full list of author information is available at the end of the article



the Maple implementation of the proposed algorithms with sample computations.

Preliminaries

In this paper, we consider the non-linear equation of the type

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

Iterations techniques are a common approach widely used in various numerical algorithms/methods. It is a hope that an iteration in the general form of $x_{n+1} = g(x_n)$ will eventually converge to the true solution α of the problem (1) at the limit when $n \rightarrow \infty$. The concern is whether this iteration will converge, and, if so, the rate of convergence. Specifically we use the following expression to represent how quickly the error $e_n = \alpha - x_n$ converges to zero. Let $e_n = \alpha - x_n$ and $e_{n+1} = \alpha - x_{n+1}$ for $n \geq 0$ be the errors at n -th and $(n + 1)$ -th iterations respectively. If two positive constants μ and p exist, and

$$\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|^p} = \frac{|\alpha - x_{n+1}|}{|\alpha - x_n|^p} = \mu, \tag{2}$$

then the sequence is said to converge to α . Here $p \geq 1$ is called the *order of convergence*, the constant μ is the *rate of convergence* or *asymptotic error constant*. This expression may be better understood when it is interpreted as $|e_{n+1}| = \mu|e_n|^p$ when $n \rightarrow \infty$. Obviously, the larger p and the smaller μ , the more quickly the sequence converges.

Theorem 1 [3] *Suppose that $\phi \in C^p[a, b]$. If $\phi^{(k)}(x) = 0$, for $k = 0, 1, 2, \dots, p - 1$ and $\phi^{(p)}(x) \neq 0$, then the sequence $\{x_n\}$ is of order p .*

This paper focus on developing iterative methods/algorithms that are having the order of converges three, six and seven respectively. The following section presents the proposed methods using Taylor’s series and modified HPT.

Main text

In this section, we present new iterative methods and its order of convergences with numerical examples, maple implementation and sample computations using maple mathematical software tool.

New iterative methods

We assume that α is an exact root of the equation (1) and let a be an initial approximation (sufficiently close) to α . We can rewrite the non-linear equation (1) using Taylor’s series expansion as coupled system

$$f(x) = f(a) + (x - a)f'(a) + \frac{(x - a)^2}{2}f''(a) + G(x) = 0 \text{ or } \tag{3}$$

$$G(x) = f(x) - f(a) - (x - a)f'(a) - \frac{(x - a)^2}{2}f''(a). \tag{4}$$

We have, from Newton’s method, that

$$\begin{aligned} x &= a - \frac{f(a)}{f'(a)} \implies x - a \\ &= -\frac{f(a)}{f'(a)} \implies (x - a)^2 = \left(-\frac{f(a)}{f'(a)}\right)^2. \end{aligned} \tag{5}$$

From (4) and (5), we have

$$G(x) = f(x) - f(a) - (x - a)f'(a) - \frac{1}{2}\left(-\frac{f(a)}{f'(a)}\right)^2 f''(a). \tag{6}$$

We can write (6) in the following form

$$x = a - \frac{f(a)}{f'(a)} - \frac{(f(a))^2 f''(a)}{2(f'(a))^3} - \frac{G(x)}{f'(a)}.$$

It can be expressed in the form of

$$x = c + T(x), \tag{7}$$

where

$$c = a - \frac{f(a)}{f'(a)} - \frac{(f(a))^2 f''(a)}{2(f'(a))^3}, \tag{8}$$

$$T(x) = -\frac{G(x)}{f'(a)}. \tag{9}$$

Here $T(x)$ is a non-linear operator. It is clear, from relation (4), that

$$G(x_0) = f(x_0). \tag{10}$$

Note that the equation (10) will play important role in the derivation of the iteration methods, see for example [2]. We use the technique of homotopy perturbation to develop the proposed iterative algorithms to solve the given non-linear equation (1). Using the HPT, we can construct a homotopy $H(v, p, m) : \mathbb{R} \times [0, 1] \times \mathbb{R} \rightarrow \mathbb{R}$ satisfying

$$H(v, p, m) = v - c - pT(v) - p(1 - p)m = 0, \tag{11}$$

where $p \in [0, 1]$ is embedding parameter and $m \in \mathbb{R}$ is unknown number. Clearly, from (11), we have

$$\begin{aligned} H(v, 0, m) &= v - c = 0, \text{ and} \\ H(v, 1, m) &= v - c - T(v) = 0. \end{aligned}$$

Hence, the parameter p is monotonically increases on $[0, 1]$. The solution of equation (11) can be expressed as a power series in p

$$v = \sum_{i=0}^{\infty} v_i p^i. \tag{12}$$

Now the approximate solution of (1) is

$$x = \lim_{p \rightarrow 1} v = \sum_{i=0}^{\infty} x_i. \tag{13}$$

One can express the equation (11), as follows, by expanding $T(x)$ using Taylor’s series expansion around x_0 ,

$$v - c - p \left[T(x_0) + (v - x_0) T'(x_0) + \frac{(v - x_0)^2}{2} T''(x_0) + \dots \right] - p(1 - p)m = 0. \tag{14}$$

By Putting (12) in (14), we get

$$\sum_{i=0}^{\infty} v_i p^i - c - p(1 - p)m - p \left[T(x_0) + \left(\sum_{i=0}^{\infty} v_i p^i - x_0 \right) T'(x_0) + \left(\sum_{i=0}^{\infty} v_i p^i - x_0 \right)^2 \frac{T''(x_0)}{2} + \dots \right] = 0. \tag{15}$$

By comparing the coefficients of powers of p , we get

$$p^0 : x_0 - c = 0 \tag{16}$$

$$p^1 : x_1 - T(x_0) - m = 0 \tag{17}$$

$$p^2 : x_2 - x_1 T'(x_0) + m = 0 \tag{18}$$

$$p^3 : x_3 - x_2 T'(x_0) - \frac{1}{2} x_1^2 T''(x_0) = 0: \tag{19}$$

From (17), we have $x_1 = T(x_0) + m$. To obtain the value of m , assume $x_2 = 0$. Now from (18)

$$m = \frac{T(x_0) T'(x_0)}{1 - T'(x_0)}. \tag{20}$$

Now, $x_0, x_1, x_2, x_3, \dots$ are obtained as follows. From (16), we have

$$x_0 = c \implies x_0 = a - \frac{f(a)}{f'(a)} - \frac{(f(a))^2 f''(a)}{2(f'(a))^3}. \tag{21}$$

From (17) and (20), we have

$$x_1 = T(x_0) + m = \frac{T(x_0)}{1 - T'(x_0)}. \tag{22}$$

From the assumption $x_2 = 0$ and from (19), we get

$$x_3 = x_2 T'(x_0) + \frac{1}{2} x_1^2 T''(x_0) = \frac{T^2(x_0) T''(x_0)}{2(1 - T'(x_0))^2}. \tag{23}$$

From (6), (10) and (9), we have

$$T(x_0) = -\frac{G(x_0)}{f'(a)} = -\frac{f(x_0)}{f'(a)}, \tag{24}$$

$$T'(x_0) = -\frac{G'(x_0)}{f'(a)} = -\frac{f'(x_0) - f'(a)}{f'(a)} = 1 - \frac{f'(x_0)}{f'(a)}, \tag{25}$$

$$T''(x_0) = -\frac{G''(x_0)}{f'(a)} = -\frac{f''(x_0)}{f'(a)}. \tag{26}$$

The approximate solution is obtained as

$$x = \lim_{i \rightarrow \infty} x_i = x_0 + x_1 + x_2 + \dots + x_i. \tag{27}$$

This formulation allows us to form the following iterative methods.

Algorithm 1 For $i = 0$, we have

$$x \approx x_0 = a - \frac{f(a)}{f'(a)} - \frac{(f(a))^2 f''(a)}{2(f'(a))^3}.$$

Hence, for a given x_0 , we have the following iterative formula to find the approximate solution x_{n+1} .

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3}. \tag{28}$$

Algorithm 2 For $i = 1$, we have

$$\begin{aligned} x \approx x_0 + x_1 &= a - \frac{f(a)}{f'(a)} - \frac{(f(a))^2 f''(a)}{2(f'(a))^3} + \frac{T(x_0)}{1 - T'(x_0)} \\ &= a - \frac{f(a)}{f'(a)} - \frac{(f(a))^2 f''(a)}{2(f'(a))^3} - \frac{f(x_0)}{f'(x_0)} \end{aligned}$$

Hence, for a given x_0 , we have the following iterative schemes to find the approximate solution x_{n+1} .

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3}, \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3} - \frac{f(y_n)}{f'(y_n)}. \end{aligned} \tag{29}$$

Note: Since $x_2 = 0$, we have the formula (29) for $i = 2$. i.e., $x \approx x_0 + x_1 = x_0 + x_1 + x_2$.

Algorithm 3 For $i = 3$, we have

$$\begin{aligned} x \approx x_0 + x_1 + x_2 + x_3 &= a - \frac{f(a)}{f'(a)} - \frac{(f(a))^2 f''(a)}{2(f'(a))^3} + \frac{T(x_0)}{1 - T'(x_0)} + \frac{T^2(x_0) T''(x_0)}{2(1 - T'(x_0))^2} \\ &= a - \frac{f(a)}{f'(a)} - \frac{(f(a))^2 f''(a)}{2(f'(a))^3} - \frac{f(x_0)}{f'(x_0)} - \frac{(f(x_0))^2 f''(x_0)}{2f'(a)(f'(x_0))^2} \end{aligned}$$

Hence, for a given x_0 , we have the following iterative formula to find the approximate solution x_{n+1} .

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3}, \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3} \\ &\quad - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^2 f''(y_n)}{2f'(x_n)(f'(y_n))^2}. \end{aligned} \tag{30}$$

Order of convergence

In this section, we show, in the following theorems, that the orders of converges of Algorithms 1, 2 and 3 are three, six and seven respectively. Let $I \subset \mathbb{R}$ be an open interval. To prove this, we follow the proofs of [9, Theorem 5, Theorem 6].

Theorem 2 Let $f : I \rightarrow \mathbb{R}$. Suppose $\alpha \in I$ is a simple root of (1) and θ is a sufficiently small neighborhood of α . Let $f''(x)$ exist and $f'(x) \neq 0$ in θ . Then the iterative formula (28) given in Algorithm 1 produces a sequence of iterations $\{x_n : n = 0, 1, 2, \dots\}$ with order of convergence three.

Proof

Let

$$R(x) = x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}.$$

Since α is a root of $f(x)$, hence $f(\alpha) = 0$. One can compute that

$$\begin{aligned} R(\alpha) &= \alpha, \\ R'(\alpha) &= 0, \\ R''(\alpha) &= 0, \\ R'''(\alpha) &\neq 0. \end{aligned}$$

Hence the Algorithm 1 has third order convergence, by Theorem 1. \square

One can also verify that the order of convergence of Algorithm 1 as in the following example.

Example 1

Consider the following equation. It has a root $\alpha = \sqrt{30}$. We show, as discussed in proof of Theorem 2, that the Algorithm 1 has third order convergence.

$$f(x) = 30 - x^2. \tag{31}$$

Following Theorem 2, we have

$$\begin{aligned} R(x) &= x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3} \\ &= \frac{3x^4 + 180x^2 - 900}{8x^3}, \\ R'(x) &= \frac{3(x^2 - 30)^2}{8x^4}, \\ R''(x) &= \frac{45(x^2 - 30)}{x^5}, \\ R'''(x) &= -\frac{135(x^2 - 50)}{x^6}. \end{aligned}$$

Now

$$\begin{aligned} R(\alpha) &= \sqrt{30} = \alpha, \\ R'(\alpha) &= 0, \\ R''(\alpha) &= 0, \\ R'''(\alpha) &= \frac{1}{10} \neq 0. \end{aligned}$$

Hence, by Theorem 2, the Algorithm 1 has third order convergence.

Theorem 3 Let $f : I \rightarrow \mathbb{R}$. Suppose $\alpha \in I$ is a simple root of (1) and θ is a sufficiently small neighborhood of α . Let $f''(x)$ exist and $f'(x) \neq 0$ in θ . Then the iterative formula (29) given in Algorithm 2 produces a sequence of iterations $\{x_n : n = 0, 1, 2, \dots\}$ with order of convergence six.

Proof

Let

$$R(x) = x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3} - \frac{f\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)}{f'\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)}$$

Since α is a root of $f(x)$, hence $f(\alpha) = 0$. One can compute that

$$\begin{aligned} R(\alpha) &= \alpha, \\ R'(\alpha) &= 0, \\ R''(\alpha) &= 0, \\ R^{(3)}(\alpha) &= 0, \\ R^{(4)}(\alpha) &= 0, \\ R^{(5)}(\alpha) &= 0, \\ R^{(6)}(\alpha) &\neq 0. \end{aligned}$$

Hence the Algorithm 2 has sixth order convergence, by Theorem 1. □

We can also verify the order of convergence of Algorithm 2 as in the following example.

Example 2

Consider the equation (31). Using Theorem 3, similar to Example 1, we have

$$\begin{aligned} R(x) &= x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3} - \frac{f\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)}{f'\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)} \\ &= \frac{3x^8 + 1000x^6 + 9000x^4 - 108000x^2 + 270000}{16x^3(x^4 + 60x^2 - 300)}, \\ R'(x) &= \frac{(3x^8 - 280x^6 + 9000x^4 - 108000x^2 + 270000)(x^2 - 30)^2}{16x^4(x^4 + 60x^2 - 300)^2}, \\ R''(x) &= \left(\frac{5(x^2 - 30)}{2x^5(x^4 + 60x^2 - 300)^3}\right) \\ &\quad (41x^{12} - 3180x^{10} + 60300x^8 + 684000x^6 \\ &\quad - 26730000x^4 + 145800000x^2 - 243000000), \\ R^{(3)}(x) &= \left(-\frac{15}{2x^6(x^4 + 60x^2 - 300)^4}\right) \\ &\quad (41x^{18} - 9810x^{16} + 488400x^{14} - 3348000x^{12} \\ &\quad - 162900000x^{10} - 955800000x^8 + 10486800000x^6 \\ &\quad - 884520000000x^4 + 2988900000000x^2 - 3645000000000), \\ R^{(4)}(x) &= \left(\frac{30}{x^7(x^4 + 60x^2 - 300)^5}\right) \\ &\quad (41x^{22} - 17175x^{20} + 1308000x^{18} - 15727500x^{16} \\ &\quad - 296100000x^{14} - 14625900000x^{12} + 284580000000x^{10} \\ &\quad + 8460450000000x^8 - 117733500000000x^6 \\ &\quad + 6506325000000000x^4 - 16621200000000000x^2 \\ &\quad + 16402500000000000), \\ R^{(5)}(x) &= \left(-\frac{150}{x^8(x^4 + 60x^2 - 300)^6}\right) \\ &\quad (41x^{26} - 26505x^{24} + 3009600x^{22} - 63693000x^{20} \\ &\quad - 95310000x^{18} - 63030150000x^{16} - 34344000000x^{14} \\ &\quad + 51666660000000x^{12} + 470472300000000x^{10} \\ &\quad - 12040285500000000x^8 + 100252080000000000x^6 \\ &\quad - 4074381000000000000x^4 + 8332470000000000000x^2 \\ &\quad - 68890500000000000000), \\ R^{(6)}(x) &= \left(\frac{900}{x^9(x^4 + 60x^2 - 300)^7}\right) \\ &\quad (41x^{30} - 37800x^{28} + 6113100x^{26} - 208782000x^{24} \\ &\quad + 1884330000x^{22} - 208202400000x^{20} - 2671893000000x^{18} \\ &\quad + 144949500000000x^{16} + 5848094700000000x^{14} \\ &\quad + 9219420000000000x^{12} - 96336135000000000x^{10} \\ &\quad + 12133913400000000000x^8 - 71049069000000000000x^6 \\ &\quad + 2277979200000000000000x^4 - 3877551000000000000000x^2 \\ &\quad + 275562000000000000000000). \end{aligned}$$

Now, we can check that

$$\begin{aligned}
 R(\alpha) &= \sqrt{30} = \alpha, \\
 R'(\alpha) &= 0, \\
 R''(\alpha) &= 0, \\
 R^{(3)}(\alpha) &= 0, \\
 R^{(4)}(\alpha) &= 0, \\
 R^{(5)}(\alpha) &= 0, \\
 R^{(6)}(\alpha) &= \frac{\sqrt{30}}{300} \neq 0.
 \end{aligned}$$

Hence, by Theorem 3, the Algorithm 2 has sixth order convergence.

Theorem 4 *Let $f : I \rightarrow \mathbb{R}$. Suppose $\alpha \in I$ is a simple root of (1) and θ is a sufficiently small neighborhood of α . Let $f''(x)$ exist and $f'(x) \neq 0$ in θ . Then the iterative formula (30) given in Algorithm 3 produces a sequence of iterations $\{x_n : n = 0, 1, 2, \dots\}$ with order of convergence seven.*

Proof
Let

Since α is a root of $f(x)$, hence $f(\alpha) = 0$. One can compute that

$$\begin{aligned}
 R(\alpha) &= \alpha, \\
 R'(\alpha) &= 0, \\
 R''(\alpha) &= 0, \\
 R^{(3)}(\alpha) &= 0, \\
 R^{(4)}(\alpha) &= 0, \\
 R^{(5)}(\alpha) &= 0, \\
 R^{(6)}(\alpha) &= 0, \\
 R^{(7)}(\alpha) &\neq 0.
 \end{aligned}$$

Hence the Algorithm 3 has seventh order convergence, by Theorem 1. □

Again, one can verify the order of convergence of Algorithm 3 using the following example.

Example 3

Consider the equation (31). Following Theorem 4, similar to Example 1 and Example 2, we have

$$\begin{aligned}
 R(x) &= x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3} - \frac{f\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)}{f'\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)} \\
 &\quad - \frac{\left(f\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)\right)^2 f''\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)}{2f'(x)\left(f'\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)\right)^2}.
 \end{aligned}$$

$$\begin{aligned}
 R(x) &= x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3} - \frac{f\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)}{f'\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)} \\
 &\quad - \frac{\left(f\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)\right)^2 f''\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)}{2f'(x)\left(f'\left(x - \frac{f(x)}{f'(x)} - \frac{(f(x))^2 f''(x)}{2(f'(x))^3}\right)\right)^2} \\
 &= \left(\frac{1}{512x^7(x^4 + 60x^2 - 300)^2}\right)(87x^{16} + 39440x^{14} + 2046800x^{12} \\
 &\quad + 9912000x^{10} - 428220000x^8 + 3650400000x^6 - 19116000000x^4 \\
 &\quad + 58320000000x^2 - 72900000000).
 \end{aligned}$$

Now, we can check that

$$\begin{aligned}
 R(\alpha) &= \sqrt{30} = \alpha, \\
 R'(\alpha) &= 0, \\
 R''(\alpha) &= 0, \\
 R^{(3)}(\alpha) &= 0, \\
 R^{(4)}(\alpha) &= 0, \\
 R^{(5)}(\alpha) &= 0, \\
 R^{(6)}(\alpha) &= 0, \\
 R^{(7)}(\alpha) &= \frac{7}{300} \neq 0.
 \end{aligned}$$

Hence, by Theorem 4, the Algorithm 3 has seventh order convergence.

Numerical example

This section presents several numerical examples to illustrate the proposed algorithms, and comparisons are made to confirm that the proposed algorithms give solution faster than existing methods.

Example 4

Consider a non-linear equation

$$x^2 - e^x - 3x + 2 = 0. \tag{32}$$

Suppose the initial approximation is $x_0 = 2$ with tolerance error 10^{-10} correct to ten decimal places. Following the proposed algorithms (in equations 28, 29 and 30), we have

$$\begin{aligned}
 x_0 &= 2, \\
 f(x) &= x^2 - e^x - 3x + 2 \text{ and } f(x_0) = -7.389056099, \\
 f'(x) &= 2x - e^x - 3 \text{ and } f'(x_0) = -6.389056099, \\
 f''(x) &= 2 - e^x \text{ and } f''(x_0) = -5.389056099.
 \end{aligned}$$

Iteration-1 using Algorithm 1:

$$\begin{aligned}
 x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{(f(x_0))^2 f''(x_0)}{2(f'(x_0))^3} \\
 &= 0.2793895885.
 \end{aligned}$$

Iteration-2 using Algorithm 1:

$$\begin{aligned}
 x_1 &= 0.2793895885, \quad f(x_1) = -0.082432628, \\
 f'(x_1) &= -3.763543228, \quad f''(x_1) = 0.677677595.
 \end{aligned}$$

Now,

$$\begin{aligned}
 x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)} - \frac{(f(x_1))^2 f''(x_1)}{2(f'(x_1))^3} \\
 &= 0.2575298491.
 \end{aligned}$$

Iteration-3 using Algorithm 1:

$$\begin{aligned}
 x_2 &= 0.2575298491, \quad f(x_2) = 0.000001649, \\
 f'(x_2) &= -3.778670729, \quad f''(x_2) = 0.706269573.
 \end{aligned}$$

Now,

$$\begin{aligned}
 x_3 &= x_2 - \frac{f(x_2)}{f'(x_2)} - \frac{(f(x_2))^2 f''(x_2)}{2(f'(x_2))^3} \\
 &= 0.2575302855.
 \end{aligned}$$

Similarly, the Iteration-4 using Algorithm 1 is $x_4 = 0.2575302855$. One can observe that Iteration-3 and Iteration-4 are same up to ten decimal places and also the tolerance error is 10^{-10} . Hence the required approximate root of the given equation (32) is 0.2575302855.

Now, we compute the iterations using Algorithm 2 as follows.

Iteration-1 using Algorithm 2:

$$\begin{aligned}
 y_0 &= x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{(f(x_0))^2 f''(x_0)}{2(f'(x_0))^3} \\
 &= 0.2793895885, \\
 x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{(f(x_0))^2 f''(x_0)}{2(f'(x_0))^3} - \frac{f(y_0)}{f'(y_0)} \\
 &= 0.2574866574.
 \end{aligned}$$

Iteration-2 using Algorithm 2:

$$\begin{aligned}
 y_1 &= x_1 - \frac{f(x_1)}{f'(x_1)} - \frac{(f(x_1))^2 f''(x_1)}{2(f'(x_1))^3} \\
 &= 0.2575302856, \\
 x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)} - \frac{(f(x_1))^2 f''(x_1)}{2(f'(x_1))^3} - \frac{f(y_1)}{f'(y_1)} \\
 &= 0.2575302853.
 \end{aligned}$$

Similarly, the Iteration-3 using Algorithm 2 is $x_3 = 0.2575302853$. One can observe that Iteration-2 and Iteration-3 are same up to ten decimal places and also the tolerance error is 10^{-10} .

Table 1 Comparing No. of iterations by different methods

Eq.	ER	NR	NM	A1	A2	A3
(a)	0.7390851332	5	3	3	2	2
(b)	0.1118325592	4	2	1	1	1
(c)	- 1.404491648	7	DIV	4	3	2
(d)	1.696812387	DIV	DIV	5	3	3
(e)	2.154434690	6	4	4	3	2

Now, the iterations using Algorithm 3 are as follows.

Iteration-1 using Algorithm 3:

$$\begin{aligned}
 y_0 &= x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{(f(x_0))^2 f''(x_0)}{2(f'(x_0))^3} \\
 &= 0.2793895885, \\
 x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{(f(x_0))^2 f''(x_0)}{2(f'(x_0))^3} - \frac{f(y_0)}{f'(y_0)} - \frac{f(y_0)}{f'(y_0)} \\
 &\quad - \frac{(f(y_0))^2 f''(y_0)}{2f'(x_0)(f'(y_0))^2} \\
 &= 0.2574612148.
 \end{aligned}$$

Iteration-2 using Algorithm 3:

$$\begin{aligned}
 y_1 &= x_1 - \frac{f(x_1)}{f'(x_1)} - \frac{(f(x_1))^2 f''(x_1)}{2(f'(x_1))^3} = 0.2575302854, \\
 x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)} - \frac{(f(x_1))^2 f''(x_1)}{2(f'(x_1))^3} - \frac{f(y_1)}{f'(y_1)} - \frac{f(y_1)}{f'(y_1)} - \frac{(f(y_1))^2 f''(y_1)}{2f'(x_1)(f'(y_1))^2} \\
 &= 0.2575302854.
 \end{aligned}$$

Example 5

Consider the following equations with corresponding initial approximations to compare results of the proposed three methods with other existing methods. We take tolerance error 10^{-15} with correct to 15 decimal places.

- (a) $\cos x - x = 0$ with initial approximation $x_0 = 1.7$,
- (b) $xe^{-x} - 0.1 = 0$ with initial approximation $x_0 = 0.1$,
- (c) $\sin^2 x - x^2 + 1 = 0$ with initial approximation $x_0 = -1$,
- (d) $x - e^{\sin x} + 1 = 0$ with initial approximation $x_0 = 4$,
- (e) $x^3 - 10 = 0$ with initial approximation $x_0 = 1.5$.

Table 1 gives a comparison of iterations number with different methods. In the table, ER, NR, NM, A1, A2, A3 and DIV indicate Exact Root, Newton-Raphson method, Noor Method [2], Algorithms 1, 2 and 3 and diverges respectively.

From Table 1, it is clear that the numerical results show that the proposed methods are more efficient than other existing methods.

Maple implementation

In this section, we present implementation of the proposed Algorithms 1, 2 and 3 in Maple. Various maple implementations for differential and transcendental equations are available, see, for example [17–27, 35]. One

can also implement the proposed algorithms in Microsoft Excel similar to the implementation of existing algorithms in [33, 34].

Pseudo code

Input: Given $f(x)$; initial approximation $x[0]$; tolerance ϵ ; correct to decimal places δ ; maximum number of iterations n .

Output: Approximate solution

- I. for i from 0 to n do
 - II. Set $x[i + 1] =$ formula (28), (29) or (30).
 - III. if $|x[i + 1] - x[i]| < \epsilon$ and $|f(x[i + 1])| < \delta$ then break; Output $x[i + 1]$

Maple code

We present the maple code of the proposed algorithms as follows, and sample computations presented in Section.

Algorithm 1 in Maple

```

> eps := type;      # epsilon value
> epsabs := type;  # delta value
> f(x) := type;    # given function;
> fd := D(f);
> fdd := D(D(f));
> x[0] := type;    # initla approximation
> n := type;      # Maximum number of iterations

> for i from 0 to n do

> x[i+1] := x[i]-f(x[i])/fd(x[i])
      -((f(x[i]))^(2)*fdd(x[i]))/(2*(fd(x[i]))^(3)));

> if abs(x[i+1]-x[i]) < eps and abs(f(x[i+1])) < epsabs then
> break;
> end if;
> printf("Iteration No: %g = %10.10f\n", i+1, x[i+1]);
> printf("f(x) = %g\n", f(x[i]));

> end do;

```

Algorithm 2 in Maple

```

> eps := type;      # epsilon value
> epsabs := type;  # delta value
> f(x) := type;    # given function
> fd := D(f);
> fdd := D(D(f));
> x[0] := type;    # initla approximation
> n := type;      # Maximum number of iterations

> for i from 0 to n do

> y[i] := x[i]-f(x[i])/fd(x[i])-f(x[i])^2*fdd(x[i])/(2*fd(x[i])^3);

> x[i+1] := x[i]-f(x[i])/fd(x[i])-f(x[i])^2*fdd(x[i])/(2*fd(x[i])^3)
      -f(y[i])/fd(y[i])

> if abs(x[i+1]-x[i]) < eps and abs(f(x[i+1])) < epsabs then
> break;
> end if;

> printf("Iteration No: %g = %10.10f\n", i+1, x[i+1]);
> printf("f(x) = %g\n", f(x[i]));
> end do;

```

Algorithm 3 in Maple

```

> eps := type;      # epsilon value
> epsabs := type;  # delta value
> f(x) := type;    # given function
> fd := D(f);
> fdd := D(D(f));
> x[0] := type;    # initla approximation
> n := type;      # Maximum number of iterations

> for i from 0 to n do

> y[i] := x[i]-f(x[i])/fd(x[i])-f(x[i])^2*fdd(x[i])/(2*fd(x[i])^3);

> x[i+1] := x[i]-f(x[i])/fd(x[i])-f(x[i])^2*fdd(x[i])/(2*fd(x[i])^3)
          -f(y[i])/fd(y[i])-f(y[i])^2*fdd(y[i])/(2*fd(x[i])*fd(y[i])^2)

> if abs(x[i+1]-x[i]) < eps and abs(f(x[i+1])) < epsabs then
> break;
> end if;

> printf("Iteration No: %g = %10.10f\n", i+1, x[i+1]);
> printf("f(x) = %g\n", f(x[i]));
> end do;

```

Sample computations

Consider the following function for sample computations using the Maple implementation.

$$f(x) = \frac{1}{7}(30 - x^2),$$

with initial approximation $x[0] = 3.5$, tolerance $\epsilon = 10^{-5}$, correct to decimal places $\delta = 10^{-10}$ (i.e., up to 10 decimal places);, and maximum number of iterations $n = 10$.

Algorithm 1 sample computations using Maple

```

> eps := 0.0000000001;
> epsabs := 0.0000000001;
> f(x) := 1/7*(30-x^2);
> fd := D(f);
> fdd := D(D(f));
> x[0] := 3.5;
> n := 10;

> for i from 0 to n do

> x[i+1] := x[i]-(f(x[i]))/(fd(x[i]))
-((f(x[i]))^(2)*fdd(x[i]))/(2*(fd(x[i]))^(3)));

> if abs(x[i+1]-x[i])<eps and abs(f(x[i+1]))<epsabs then
> break;
> end if;

> printf("Iteration No: %g = %10.10f\n", i+1, x[i+1]);
> printf("f(x) = %g\n", f(x[i]));

> end do;
1.10-10
1.10-10
 $x \rightarrow \frac{30}{7} - \frac{1}{7}x^2$ 
 $x \rightarrow -\frac{2}{7}x$ 
 $x \rightarrow -\frac{2}{7}$ 
3.5
10

5.117164723
> Iteration No: 1 = 5.1171647230
f(x) = 2.53571
5.476318565
> Iteration No: 2 = 5.4763185650
f(x) = 0.544946
5.477225575
> Iteration No: 3 = 5.4772255750
f(x) = 0.00141928
5.477225575

```

$$1.10^{-10}$$
$$1.10^{-10}$$
$$x \rightarrow \frac{30}{7} - \frac{1}{7}x^2$$
$$x \rightarrow -\frac{2}{7}x$$
$$x \rightarrow -\frac{2}{7}$$
$$3.5$$
$$10$$
$$5.117164723$$

> Iteration No: 1 = 5.1171647230
f(x) = 2.53571
5.476318565

> Iteration No: 2 = 5.4763185650
f(x) = 0.544946
5.477225575

> Iteration No: 3 = 5.4772255750
f(x) = 0.00141928
5.477225575

Algorithm 2 sample computations using Maple

```

> eps := 0.0000000001;
> epsabs := 0.0000000001;
> f(x) := 1/7*(30-x^2);
> fd := D(f);
> fdd := D(D(f));
> x[0] := 3.5;
> n := 10;

> for i from 0 to n do
> y[i] := x[i]-f(x[i])/fd(x[i])-f(x[i])^2*fdd(x[i])/(2*fd(x[i])^3);

> x[i+1] := x[i]-f(x[i])/fd(x[i])-f(x[i])^2*fdd(x[i])/(2*fd(x[i])^3)
-f(y[i])/fd(y[i])

> if abs(x[i+1]-x[i])<eps and abs(f(x[i+1]))<epsabs then
> break;
> end if;

> printf("Iteration No: %g = %10.10f\n", i+1, x[i+1]);
> printf("f(x) = %g\n", f(x[i]));
> end do;

```

$$1.10^{-10}$$

$$1.10^{-10}$$

$$x \rightarrow \frac{30}{7} - \frac{1}{7}x^2$$

$$x \rightarrow -\frac{2}{7}x$$

$$x \rightarrow -\frac{2}{7}$$

$$3.5$$

$$10$$

$$5.117164723$$

$$5.489893119$$

> Iteration No: 1 = 5.4898931190

$$f(x) = 2.53571$$

$$5.477225609$$

$$5.477225575$$

> Iteration No: 2 = 5.4772255750

$$f(x) = -0.0198466$$

$$5.477225575$$

$$5.477225575$$

```

1.10-10
1.10-10
x →  $\frac{30}{7} - \frac{1}{7}x^2$ 
x →  $-\frac{2}{7}x$ 
x →  $-\frac{2}{7}$ 
3.5
10
5.117164723
5.489893119
> Iteration No: 1 = 5.4898931190
f(x) = 2.53571
5.477225609
5.477225575
> Iteration No: 2 = 5.4772255750
f(x) = -0.0198466
5.477225575
5.477225575

```

Similarly, one can apply the Algorithm 3 using Maple code.

Conclusion

In this paper, we presented three iterative methods of order three, six and seven respectively for solving non-linear equations. With the help of modified homotopy perturbation technique, we obtained coupled system of equations which gives solution faster than existing methods. The analysis of convergence of the proposed iterative methods are discussed with example for each proposed method. Maple implementations of the proposed methods are discussed with sample sample computations. Numerical examples are presented to illustrate and validation of the proposed methods.

Limitations

The proposed algorithms are implemented in Maple only. However, we can also implement these algorithms in Mathematica, SCILab, Matlab, Microsoft Excel etc.

Abbreviation

HPT: Homotopy Perturbation Technique.

Acknowledgements

The authors are thankful to the reviewers and editor for providing valuable inputs to improve the quality and present format of manuscript.

Author contributions

ST involved in creation of the proposed algorithms for solving non-linear equations using the modified HPT, the convergence analysis, and Maple implementation. PS is involved in suggestion and verification of the numerical examples in the present paper. Both authors read and approved the final manuscript.

Authors' information

Prof. Srinivasarao Thota completed his M.Sc. in Mathematics from Indian Institute of Technology (IIT) Madras, India and Ph.D. in Mathematics from Motilal Nehru National institute of Technology (NIT) Allahabad, India. Prof. Thota's area of research interests are Computer Algebra (symbolic methods for differential equations), Numerical Analysis (root finding algorithms), Mathematical Modeling (ecology). He has published more than 40 research papers in various international journals and presented his research work at several international conferences as oral presenter and invited/keynote/guest speaker in different countries. Presently working at department of mathematics, SR University, Warangal, India.

Prof. P. Shanmugasundaram completed M.Sc. in Mathematics from Sri Vasavi College, M.Phil in Mathematics from Madurai Kamraj University, Madurai, India and Ph.D. in Mathematics Anna University, Chennai, India. He has published more than 25 research papers in various international journals. Presently, working at Department of Mathematics, College of Natural & Computational sciences, Mizan-Tepi University, Ethiopia.

Funding

Not applicable.

Availability of data and materials

The datasets generated and analyzed during the current study are presented in this manuscript.

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Consent for publication

Not applicable.

Author details

¹Department of Mathematics, School of Sciences, SR University, Warangal, Telangana 506371, India. ²Department of Mathematics, College of Natural & Computational sciences, Mizan Tepi University, Mizan Teferi, Ethiopia.

Received: 23 May 2022 Accepted: 12 July 2022

Published online: 30 July 2022

References

- Naseem A, Rehman MA, Abdeljawad T. Real-world applications of a newly designed root-finding algorithm and its polynomiography. *IEEE Access*. 2021;9:160868–77.
- Chun C. Iterative methods improving Newton's method by the decomposition method. *Comput Math Appl*. 2005;50:1559–68.
- Babolian E, Biazar J. On the order of convergence of adomain method. *Appl Math Comput*. 2002;130:383–7.
- He JH. A new iterative method for solving algebraic equations. *Appl Math Comput*. 2005;135:81–4.
- Noor MA, Noor KI, Khan WA, Ahmad F. On iterative methods for nonlinear equations. *Appl Math Comput*. 2006;183:128–33.
- Noor MA, Ahmad F. Numerical comparison of iterative methods for solving nonlinear equations. *Appl Math Comput*. 2006;180:167–72.
- Noor MA, Noor KI. Three-step iterative methods for nonlinear equations. *Appl Math Comput*. 2006;183:322–7.
- Noor MA. Iterative methods for nonlinear equations using homotopy perturbation technique. *Appl Math Inf Sci*. 2010;4(2):227–35.
- Saqib M, Iqbal M, Ali S, Ismaeel T. New fourth and fifth-order iterative methods for solving nonlinear equations. *Appl Math*. 2015;6:1220–7.
- Saied L. A new modification of False-Position method based on homotopy analysis method. *Appl Math Mech*. 2008;29(2):223–8.
- Sagraloff M. Computing real roots of real polynomials. *J Symb Comput*. 2013. <https://doi.org/10.1016/j.jsc.2015.03.004>.
- Hussain S, Srivastav VK, Thota S. Assessment of interpolation methods for solving the real life problem. *Int J Math Sci Appl*. 2015;5(1):91–5.
- Thota S, Gemechu T, Shanmugasundaram P. New algorithms for computing non-linear equations using exponential series. *Palestine J Math*. 2021;10(1):128–34.
- Thota S, Srivastav VK. Quadratically convergent algorithm for computing real root of non-linear transcendental equations. *BMC Res Notes*. 2018;11:909.
- Thota S, Srivastav VK. Interpolation based hybrid algorithm for computing real root of non-linear transcendental functions. *IJMCR*. 2014;2(11):729–35.
- Thota S. A new root-finding algorithm using exponential series. *Ural Math J*. 2019;5(1):83–90.
- Thota S, Kumar SD. Maple implementation of a reduction algorithm for differential-algebraic systems with variable coefficients, International Conference on Recent Trends in Engineering & Technology and Quest for Sustainable Development, 2019, 1–8.
- Thota S, Kumar SD. Maple implementation of symbolic methods for initial value problems. *Res Resurgence*. 2018;1(1):21–39 (ISBN: 978-93-83861-12-5).
- Thota S. Initial value problems for system of differential-algebraic equations in Maple. *BMC Res Notes*. 2018;11:651.
- Thota S. A symbolic algorithm for polynomial interpolation with stieljes conditions in maple. *Proc Inst Appl Math*. 2019;8(2):112–20.
- Thota S. Maple implementation of a symbolic method for fully inhomogeneous boundary value problems. *Int J Comput Sci Eng*. 2021;3:1–5.
- Thota S, Kumar SD. A new reduction algorithm for differential-algebraic systems with power series coefficients. *Inf Sci Lett*. 2021;10(1):59–66.
- Thota S. Maple implementation for reducing differential-algebraic systems, Conference on Mathematics for the Advancement of Science, Technology and Education, March 5–6, 2021, at Ethiopian Mathematics Professionals Association (EMPA), Department of Mathematics, College of Natural and Computational Sciences, Addis Ababa University, Ethiopia.
- Thota S. Implementation of a symbolic method for fully inhomogeneous boundary value problems, 2021 International Conference on Mathematics and Computers in Science and Engineering (MACISE 2021), January 18–20, 2021, Madrid, Spain.
- Thota S. On a third order iterative algorithm for solving non-linear equations with maple implementation, National E-Conference on Interdisciplinary Research in Science and Technology, May 30–31, 2020, Amiruddaula Islmia Degree College, Locknow, India.
- Thota S, Kumar SD. Maple implementation of a reduction algorithm for differential-algebraic systems with variable coefficients, International Conference on Recent Trends in Engineering & Technology and Quest for Sustainable Development, May 13–14, 2019, Institute of Technology, Ambo University, Ethiopia.
- Thota S, Kumar SD. Maple package of initial value problem for system of differential-algebraic equations. National Seminar on Applications of Scientific and Statistical Software in Research, 30–31 March. The School of Sciences. Allahabad, India: Uttar Pradesh Rajarshi Tandon Open University; 2017.
- Gemechu T, Thota S. On new root finding algorithms for solving nonlinear transcendental equations. *Int J Chem Math Phys*. 2020;4(2):18–24.
- Thota S. Solution of generalized abel's integral equations by homotopy perturbation method with adaptation in laplace transformation. *Sohag J Math*. 2022;9(2):29–35.
- Thota S, Srivastav VK. An algorithm to compute real root of transcendental equations using hyperbolic tangent function. *Int J Open Problems Comput Math*. 2021;14(2):1–14.
- Thota S. A numerical algorithm to find a root of non-linear equations using householders method. *Int J Adv Appl Sci*. 2021;10(2):141–8.
- Thota S. Solution of generalized abel's integral equations by homotopy perturbation method with adaptation in laplace transformation. *Sohag J Math*. 2022;9(2):29–35.
- Thota S, Ayoade AA. On solving transcendental equations using various root finding algorithms with microsoft excel, 1st edition, Notion Press, 2022, ISBN-13: 979-8886844238.
- Thota S. Microsoft Excel Implementation of Numerical Algorithms for Nonlinear algebraic or Transcendental Equations, 5th International Conference on Statistics, Mathematical Modelling and Analysis (SMMA 2022), May 27–29, 2022 in Xi'an, China.
- Thota S. An introduction to maple, five days national faculty development program on innovative tools for solving research problems, April 25–29. India: Chitkara University; 2022.
- Thota S. A new hybrid halley-false position type root finding algorithm to solve transcendental equations, Istanbul International Modern Scientific Research Congress-III, 06-08. Istanbul Gedik University. Turkey: Istanbul; 2022.
- Parveen T, Singh S, Thota S, Srivastav VK. A new hydride root-finding algorithm for transcendental equations using bisection, regula-falsi and newton-raphson methods, National Conference on Sustainable & Recent Innovation in Science and Engineering (SUNRISE-19), 2019. (ISBN No. 978-93-5391-715-9).
- Srivastav VK, Thota S, Kumar M. A new trigonometrical algorithm for computing real root of non-linear transcendental equations. *Int J Appl Comput Math*. 2019;5:44.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.