Tamap Journal of Mathematics and Statistics
http://www.tamap.org/
doi:10.29371/2018.16.24
Volume 2018, Article ID 24
Research Article

A Modification of Newton's Method Using Gauss-Legendre Quadrature Rules

Mohammad Masjed-Jamei[1] and Zahra Moalemi[1,*]

[1] Department of Mathematics, K.N.Toosi University of Technology, P.O. Box 16315-1618, Tehran, Iran.

Received: 11.12.2017 • **Accepted**: 05.03.2018 • **Published Online:** 06.03.2018

Abstract: In this paper, a fourth-order modification of Newton's method, which is based on Gauss-Legendre quadrature rules, is constructed for solving nonlinear equations. The convergence analysis of the proposed method is studied and numerical examples are given to compare the efficiency of the method with respect to other known iterative methods.

Keywords: Newton's Method; Iterative methods; Gauss-Legendre quadrature rule; Order of convergence; Asymptotic error constant.

1.Introduction

Let us consider the problem of finding a simple root, say α , of the nonlinear equation

$$f(x)=0, \tag{0.1}$$

where $f:D\subseteq\mathbb{R}\to\mathbb{R}$, for an open interval D, is a scalar function and is sufficiently differentiable in a neighborhood of α . Various iterative methods have been applied up to now for solving the problem (1.1) by using different techniques such as quadrature formulas, Taylor series, decomposition techniques and homotopy perturbation method [7]. Newton's method is one of the most applicable numerical techniques to solve nonlinear equation (1.1) whose iterative scheme is as

$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}, \tag{0.2}$$

which is quadratically convergent in the neighborhood of α if and only if $f(\alpha)=0$ and $f'(\alpha)\neq 0$.

Definition 1.1. [19] If the sequence $\{x_n\}$ tends to a limit α such that

$$\lim_{n\to\infty}\frac{x_{n+1}-\alpha}{(x_n-\alpha)^p}=C,$$

for some $C\neq 0$ and $p\geq 1$, then p is called the order of convergence, and C is the asymptotic error constant (AEC).

For $p=1$, $p=2$ or $p=3$, the sequence is said to converge linearly, quadratically or cubically, respectively.

Definition 1.2. [12] Let $e_n=x_n-\alpha$ be the partial error in the n th iterate of the method. Then the relation

$$e_{n+1}=Ce_n^p+O(e_n^{p+1}),$$

* Correspondence: zmoalemi@mail.kntu.ac.ir

is called the error equation where p is called the order and C is known as the asymptotic error constant of the method.

Definition 1.3. [20] Let α be a root of the function f and x_{n+1} , x_n and x_{n-1} be three consecutive iterations closer to the root α .Then the ratio

$$\frac{\ln|(x_{n+1}-\alpha)/(x_n-\alpha)|}{\ln|(x_n-\alpha)/(x_{n-1}-\alpha)|}, \tag{0.3}$$

gives the approximation of the computational order of convergence (COC).

There are some modifications of Newton's method with cubic convergence which do not contain second derivatives, see e.g. [4,5] and references therein. One of these third order methods was introduced in [20] where the trapezoidal approximation was applied to derive a modification of Newton's method with a cubically convergence order as

$$x_{n+1} = x_n - \frac{2f(x_n)}{f'(x_n)+f'(v_{n+1})}, \tag{0.4}$$

in which

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

In contrast to the Newton method, (0.5) uses the arithmetic mean of $f'(x_n)$ and $f'(v_{n+1})$, instead of $f'(x_n)$. Hence, it is called Arithmetic mean Newton's method (AN) in [16].

Frontini and Sormani [8] similarly considered the midpoint approximation to obtain a third-order method as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(\frac{x_n+v_{n+1}}{2})}.$$

Of course, this method has been independently derived by Homeier [9] as

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

Homeier also derived the following cubically convergent iteration scheme [10]

$$x_{n+1} = x_n - \frac{f(x_n)}{2}\left(\frac{1}{f'(x_n)}+\frac{1}{f'(v_{n+1})}\right), \tag{0.6}$$

by applying Newton's theorem to the inverse function $x = f(y)$ instead of $y = f(x)$ .The scheme (1.5) has been called in [16] the Harmonic mean Newton's method (HN).

Finally, if the geometric mean is used instead of arithmetic mean in (1.4), the Newton scheme changes to

$$x_{n+1} = x_n - \frac{f(x_n)}{\text{sign}(f'(x_0))\sqrt{f'(x_n)f'(v_{n+1})}}. \tag{0.7}$$

This method is called in [12] Geometric mean Newton's method (GN).

Motivated by the recent results in this area and the above-mentioned iterative methods based upon quadrature rules, in this paper we introduce a fourth-order modification of Newton's method in which a Gauss-Legendre quadrature is used.

2. GAUSS-LEGENDRE QUADRATURES

A general n -point quadrature formula is denoted by

$$\int_a^b f(x)dx = \sum_{k=0}^{n} w_{k,n} f(x_{k,n}) + R_n[f], \tag{1.1}$$

where $\{x_{k,n}\}_{k=1}^n$ and $\{w_{k,n}\}_{k=1}^n$ are respectively nodes and weight coefficients and $R_n[f]$ is the corresponding error [13].

Let $\mathbf{\Pi}_d$ be the set of algebraic polynomials of degree at most d .The quadrature formula (2.1) has degree of exactness d if for every $p \in \mathbf{\Pi}_d$ we have $R_n[p] = 0$. In addition, if $R_n[p] \neq 0$ for some $p \in \mathbf{\Pi}_{d+1}$, formula (2.1) has precise degree of exactness d . The convergence order of quadrature rule (2.1) depends on the smoothness of the function f as well as on its degree of exactness.

It is well known that polynomial interpolation can be employed in constructing numerical methods for quadrature rules, differential equations and related problems; see e.g. [15]. For given mutually different nodes $\{x_{k,n}\}_{k=1}^n$, we can always achieve a degree of exactness $d = n-1$ by interpolating at these nodes and integrating the interpolated polynomial instead of f . Namely, taking the node polynomial

$$\Psi_n(x) = \prod_{k=1}^{n} (x - x_{k,n}),$$

by integrating the Lagrange interpolation formula

$$f(x) = \sum_{k=1}^{n} f(x_{k,n}) L(x; x_{k,n}) + r_n(f; x),$$

where

$$L(x; x_{k,n}) = \frac{\Psi_n(x)}{\Psi'_n(x_{k,n})(x - x_{k,n})}, \qquad (k = 1, 2, ..., n),$$

we obtain (2.1), with

$$w_{k,n} = \frac{1}{\Psi'_n(x_{k,n})} \int_a^b \frac{\Psi_n(x)}{x - x_{k,n}} dx, \qquad (k = 1, 2, ..., n),$$

and

$$R_n[f] = \int_a^b r_n(f; x) dx.$$

It is clear that if $f \in \mathbf{\Pi}_{n-1}$ then $r_n(f; x) = 0$ and therefore $R_n[f] = 0$.

One of the important cases of interpolatory quadrature rules is the Gauss-Legendre quadrature rule

$$\int_{-1}^{1} f(x)dx = \sum_{k=1}^{n} \frac{2}{(1 - x_{k,n}^2)(P'_n(x_{k,n}))^2} f(x_{k,n}) + \frac{2^{2n+1}(n!)^4}{(2n+1)((2n)!)^3} f^{(2n)}(\xi), \qquad (-1 < \xi < 1),$$

where

$$P_n(x) = \frac{1}{2^n} \sum_{k=1}^{[n/2]} (-1)^k \binom{n}{k} \binom{2n-2k}{n} x^{n-2k},$$

is the n th degree Legendre polynomial [1] which is orthogonal with respect to the constant weight function and has n real and distinct zeros $\{x_{k,n}\}_{k=1}^n$ arranged in increasing order on $[-1,1]$.

For instance, the two-point Gauss-Legendre rule is denoted by

$$\int_{-1}^{1} f(x)dx = f(-\frac{\sqrt{3}}{3}) + f(\frac{\sqrt{3}}{3}) + \frac{1}{135} f^{(4)}(\xi), \qquad \xi \in (-1,1). \tag{1.2}$$

From (2.2) one can easily derive a general two point formula for any other interval, say $[a,b]$, by the linear change of variable

$$t=\frac{b-a}{2}x+\frac{a+b}{2}\quad\in[a,b],$$

as follows

$$\int_a^b f(x)dx=\frac{b-a}{2}\left(f\Big(\frac{3+\sqrt{3}}{6}a+\frac{3-\sqrt{3}}{6}b\Big)+f\Big(\frac{3-\sqrt{3}}{6}a+\frac{3+\sqrt{3}}{6}b\Big)\right)+\frac{1}{4320}(b-a)^5 f^{(4)}(\xi),\quad (1.3)$$

for $\xi\in(a,b)$. The error term of (2.3) shows that the two-point Gauss-Legendre quadrature is more accurate than trapezoidal rule with the error $-\frac{(b-a)^3}{12}f''(\xi)$ for $\xi\in(a,b)$ for smooth integrand functions [17], regardless of the size of $|b-a|$ and $|f^{(n)}(\xi)|$ for $n=2,4$. In the next section, we use this point to construct our iterative method. For some results on numerical improvement of Gauss-Legendre quadrature rules see e.g. [3] and for more details on error bounds for Gaussian quadrature rules, we refer [14].

3. A NEW ITERATIVE METHOD

To introduce our algorithm, first let us explain how the iterative formula (1.2) is obtained. Let α be a simple root of nonlinear equation $f(x)=0$, i.e. $f(\alpha)=0$ and $f'(\alpha)\neq 0$, where f is a sufficiently differentiable function. It is clear that we have

$$f(x)=f(x_n)+\int_{x_n}^{x} f'(t)dt.\quad (3.1)$$

If the integral in (3.1) is approximated by the midpoint rule as $(x-x_n)f'(x_n)$, then (3.1) changes to

$$f(x)\approx f(x_n)+(x-x_n)f'(x_n).\quad (3.2)$$

By replacing $x=x_{n+1}$ in (3.2) and noting this fact that $f(x_{n+1})\approx 0$, the new approximation to α is given by

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

which is the same as classical Newton's method.

Motivated by such an approach, Weerakoon and Fernando in [20] approximate the definite integral in (3.1) with the trapezoidal rule

$$\int_{x_n}^{x} f'(t)dt\approx\frac{1}{2}(x-x_n)\big(f'(x_n)+f'(x)\big),\quad (3.3)$$

to obtain

$$x_{n+1}=x_n-\frac{2f(x_n)}{f'(x_n)+f'(v_{n+1})},\quad (3.4)$$

where

$$v_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)},\qquad n=0,1,\dots\ .$$

Now, let us approximate the definite integral in (3.1) with the Gauss-Legendre quadrature (2.3), which is more accurate than trapezoidal and midpoint rules, as

$$\int_{x_n}^{x} f'(t)dt \approx \frac{1}{2}(x-x_n)\Big(f'(\frac{3+\sqrt{3}}{6}x_n+\frac{3-\sqrt{3}}{6}x)+f'(\frac{3-\sqrt{3}}{6}x_n+\frac{3+\sqrt{3}}{6}x)\Big),$$

to obtain

$$f(x) \approx f(x_n)+\frac{1}{2}(x-x_n)\Big(f'(\frac{3+\sqrt{3}}{6}x_n+\frac{3-\sqrt{3}}{6}x)+f'(\frac{3-\sqrt{3}}{6}x_n+\frac{3+\sqrt{3}}{6}x)\Big).$$

By defining

$$F(x) = f(x_n)+\frac{1}{2}(x-x_n)\Big(f'(\frac{3+\sqrt{3}}{6}x_n+\frac{3-\sqrt{3}}{6}x)+f'(\frac{3-\sqrt{3}}{6}x_n+\frac{3+\sqrt{3}}{6}x)\Big),$$

we take the next iteration as the root of $F(x)=0$. Therefore, from

$$F(x_{n+1})=0,$$

we get

$$x_{n+1}=x_n-\frac{2f(x_n)}{f'\left(\frac{3+\sqrt{3}}{6}x_n+\frac{3-\sqrt{3}}{6}x_{n+1}\right)+f'\left(\frac{3-\sqrt{3}}{6}x_n+\frac{3+\sqrt{3}}{6}x_{n+1}\right)}. \tag{3.5}$$

Relation (3.5) is an implicit scheme that requires the first derivative of the function at $(n+1)$ th iterative step to obtain the $(n+1)$ th iteration. To solve this problem we can use the iterative step (3.4) on the right hand side of (3.5) to finally obtain our algorithm as

$$x_{n+1}=x_n-\frac{2f(x_n)}{f'\left(\frac{3+\sqrt{3}}{6}x_n+\frac{3-\sqrt{3}}{6}u_{n+1}\right)+f'\left(\frac{3-\sqrt{3}}{6}x_n+\frac{3+\sqrt{3}}{6}u_{n+1}\right)},$$
$$\text{where} \quad u_{n+1}=x_n-\frac{2f(x_n)}{f'(x_n)+f'(v_{n+1})} \quad \text{and} \quad v_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}. \tag{3.6}$$

4. CONVERGENCE ANALYSIS OF THE PROPOSED ITERATIVE METHOD

Theorem 4.1. Let $f: D\subseteq \mathbb{R}\to\mathbb{R}$, where D is an open interval, be sufficiently differentiable function on the interval D and has a simple root $\alpha\in D$. If x_0 is sufficiently close to α, the convergence order of the proposed method (3.6) is four and the corresponding error equation is as

$$e_{n+1}=\Big(\frac{1}{2}c_2c_3+c_2^3\Big)e_n^4+O(e_n^5),$$

in which $e_n=x_n-\alpha$ and $c_j=\dfrac{f^{(j)}(\alpha)}{j!f'(\alpha)}$ for $j=1,2,...,5$.

Proof. Let α be a simple root of $f(x)$. First, by the Taylor expansion of $f(x_n)$ about α we have

$$\begin{aligned} f(x_n) &= f(\alpha+e_n) \\ &= f(\alpha)+f'(\alpha)e_n+\frac{1}{2!}f''(\alpha)e_n^2+\frac{1}{3!}f^{(3)}(\alpha)e_n^3+\frac{1}{4!}f^{(4)}(\alpha)e_n^4+O(e_n^5) \\ &= f'(\alpha)\big(e_n+c_2e_n^2+c_3e_n^3+c_4e_n^4+O(e_n^5)\big), \end{aligned} \tag{4.1}$$

where $c_j=\dfrac{f^{(j)}(\alpha)}{j!f'(\alpha)}$. Similarly

$$f'(x_n)=f'(\alpha)\big(1+2c_2e_n+3c_3e_n^2+4c_4e_n^3+O(e_n^4)\big). \tag{4.2}$$

By dividing (4.1) to (4.2) one gets

$$\frac{f(x_n)}{f'(x_n)} = \left(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)\right)\left(1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + O(e_n^4)\right)^{-1}$$
$$= \left(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)\right)\times$$
$$\left(1 - \left(2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + O(e_n^4)\right) + \left(2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + O(e_n^4)\right)^2 - \left(2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + O(e_n^4)\right)^3 + O(e_n^4)\right)$$
$$= \left(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)\right)\left(1 - 2c_2e_n + (4c_2^2 - 3c_3)e_n^2 + (12c_2c_3 - 8c_2^3 - 4c_4)e_n^3 + O(e_n^4)\right)$$
$$= e_n - c_2e_n^2 + (2c_2^2 - 2c_3)e_n^3 + (7c_2c_3 - 4c_2^3 - 3c_4)e_n^4 + O(e_n^5).$$

Therefore

$$v_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = \alpha + c_2e_n^2 + (2c_3 - 2c_2^2)e_n^3 + (3c_4 + 4c_2^3 - 7c_2c_3)e_n^4 + O(e_n^5). \tag{4.3}$$

Again by expanding $f'(v_{n+1})$ about α and noting (4.3) one gets

$$f'(v_{n+1})$$
$$= f'(\alpha) + \left(c_2e_n^2 + (2c_3 - 2c_2^2)e_n^3 + (3c_4 + 4c_2^3 - 7c_2c_3)e_n^4 + O(e_n^5)\right) f''(\alpha) + \frac{1}{2!}c_2^2e_n^4 f^{(3)}(\alpha) + O(e_n^5)$$
$$= f'(\alpha)\left(1 + \left(2c_2e_n^2 + 2(2c_3 - 2c_2^2)e_n^3 + 2(3c_4 + 4c_2^3 - 7c_2c_3)e_n^4 + O(e_n^5)\right)\frac{f''(\alpha)}{2f'(\alpha)} + 3c_2^2e_n^4\frac{f^{(3)}(\alpha)}{6f'(\alpha)} + O(e_n^5)\right)$$
$$= f'(\alpha)\left(1 + 2c_2^2e_n^2 + 4c_2(c_3 - c_2^2)e_n^3 + (2c_2(3c_4 + 4c_2^3 - 7c_2c_3) + 3c_3c_2^2)e_n^4 + O(e_n^5)\right). \tag{4.4}$$

Hence, sum of (4.2) and (4.4) yields

$$f'(x_n) + f'(v_{n+1}) = 2f'(\alpha)\left(1 + c_2e_n + (\frac{3}{2}c_3 + c_2^2)e_n^2 + 2(c_4 + c_2(c_3 - c_2^2))e_n^3 + O(e_n^4)\right). \tag{4.5}$$

In the sequel, referring to equations (4.1) and (4.5) yields

$$\frac{2f(x_n)}{f'(x_n) + f'(v_{n+1})}$$
$$= \left(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)\right)\left(1 + c_2e_n + (\frac{3}{2}c_3 + c_2^2)e_n^2 + 2(c_4 + c_2(c_3 - c_2^2))e_n^3 + O(e_n^4)\right)^{-1}$$
$$= \left(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)\right)\times$$
$$\Big(1 - \left(c_2e_n + (\frac{3}{2}c_3 + c_2^2)e_n^2 + 2(c_4 + c_2(c_3 - c_2^2))e_n^3 + O(e_n^4)\right)$$
$$+ \left(c_2e_n + (\frac{3}{2}c_3 + c_2^2)e_n^2 + 2(c_4 + c_2(c_3 - c_2^2))e_n^3 + O(e_n^4)\right)^2$$
$$- \left(c_2e_n + (\frac{3}{2}c_3 + c_2^2)e_n^2 + 2(c_4 + c_2(c_3 - c_2^2))e_n^3 + O(e_n^4)\right)^3 + O(e_n^4)\Big)$$
$$= \left(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)\right)\left(1 - c_2e_n - \frac{3}{2}c_3e_n^2 + (3c_2^3 + c_2c_3 - 2c_4)e_n^3 + O(e_n^4)\right)$$
$$= e_n - (\frac{1}{2}c_3 + c_2^2)e_n^3 + (3c_2^3 - \frac{3}{2}c_2c_3 - c_4)e_n^4 + O(e_n^5).$$

Thus

$$\begin{aligned} u_{n+1} &= x_n - \frac{2f(x_n)}{f'(x_n)+f'(v_{n+1})} \\ &= e_n + \alpha - \left(e_n - (\frac{1}{2}c_3 + c_2^2)e_n^3 + (3c_2^3 - \frac{3}{2}c_2c_3 - c_4)e_n^4 + O(e_n^5) \right) \\ &= \alpha + (\frac{1}{2}c_3 + c_2^2)e_n^3 + (c_4 + \frac{3}{2}c_2c_3 - 3c_2^3)e_n^4 + O(e_n^5). \end{aligned} \tag{4.6}$$

Let us define

$$X_1 := \Big(\frac{3+\sqrt{3}}{6}\Big)x_n + \Big(\frac{3-\sqrt{3}}{6}\Big)u_{n+1}, \quad X_2 := \Big(\frac{3-\sqrt{3}}{6}\Big)x_n + \Big(\frac{3+\sqrt{3}}{6}\Big)u_{n+1}.$$

According to (4.6), we can write

$$X_1 = \alpha + \Big(\frac{3+\sqrt{3}}{6}\Big)e_n + \Big(\frac{3-\sqrt{3}}{6}\Big)\Big(c_2^2 + \frac{1}{2}c_3\Big)e_n^3 + \Big(\frac{3-\sqrt{3}}{6}\Big)\Big(c_4 + \frac{3}{2}c_2c_3 - 3c_2^3\Big)e_n^4 + O(e_n^5). \tag{4.7}$$

Similarly for X_2 we have

$$X_2 = \alpha + \Big(\frac{3-\sqrt{3}}{6}\Big)e_n + \Big(\frac{3+\sqrt{3}}{6}\Big)\Big(c_2^2 + \frac{1}{2}c_3\Big)e_n^3 + \Big(\frac{3+\sqrt{3}}{6}\Big)\Big(c_4 + \frac{3}{2}c_2c_3 - 3c_2^3\Big)e_n^4 + O(e_n^5). \tag{4.8}$$

By expanding $f'(X_1)$ and $f'(X_2)$ about α using (4.7) and (4.8), we can finally obtain

$$f'(X_1) + f'(X_2) = 2f'(\alpha)\Big(1 + c_2e_n + c_3e_n^2 + R_1e_n^3 + R_2e_n^4 + O(e_n^5)\Big),$$

where $R_1 = c_4 + \frac{1}{2}c_2c_3 + c_2^3$ and $R_2 = \frac{35}{36}c_5 + \frac{1}{2}c_3^2 + c_2c_4 + \frac{5}{2}c_2^2c_3 - 3c_2^4$.

Therefore we have

$$\begin{aligned} &\frac{2f(x_n)}{f'(X_1)+f'(X_2)} \\ &= \Big(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)\Big)\Big(1 + c_2e_n + c_3e_n^2 + R_1e_n^3 + R_2e_n^4 + O(e_n^5)\Big)^{-1} \\ &= \Big(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)\Big)\times \\ &\quad \Big(1 - \Big(c_2e_n + c_3e_n^2 + R_1e_n^3 + R_2e_n^4 + O(e_n^5)\Big) + \Big(c_2e_n + c_3e_n^2 + R_1e_n^3 + R_2e_n^4 + O(e_n^5)\Big)^2 \\ &\qquad - \Big(c_2e_n + c_3e_n^2 + R_1e_n^3 + R_2e_n^4 + O(e_n^5)\Big)^3 + \Big(c_2e_n + c_3e_n^2 + R_1e_n^3 + R_2e_n^4 + O(e_n^5)\Big)^4 + O(e_n^5)\Big) \\ &= \Big(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)\Big)\times \\ &\quad \left(1 - c_2e_n + (c_2^2 - c_3)e_n^2 + (\frac{3}{2}c_2c_3 - 2c_2^3 - c_4)e_n^3 + (6c_2^4 - \frac{9}{2}c_2^2c_3 + c_2c_4 + \frac{1}{2}c_3^2 - \frac{35}{36}c_5)e_n^4 + O(e_n^5)\right) \\ &= e_n - c_2(\frac{1}{2}c_3 + c_2^2)e_n^4 + O(e_n^5). \end{aligned} \tag{4.9}$$

Finally, replacing (4.9) in the proposed scheme

$$x_{n+1} = x_n - \frac{2f(x_n)}{f'(X_1)+f'(X_2)},$$

gives

$$e_{n+1}+\alpha=e_n+\alpha-\Big(e_n-c_2\Big(\frac{1}{2}c_3+c_2^2\Big)e_n^4+O(e_n^5)\Big),$$

which is equivalent to

$$e_{n+1}=\Big(\frac{1}{2}c_2c_3+c_2^3\Big)e_n^4+O(e_n^5). \tag{4.10}$$

The latter result (4.10) approves at least a fourth-order convergence of the proposed method. ■

Note that using three point Gauss-Legendre rule (or more points) has no affection on the order of convergence and just the volume of computations increases [8].

One may now ask why we have not used the same as trapezoidal rule (3.3) instead of Gauss-Legendre rule proposed in (3.6)? In response to this question, we prove in the next theorem that using trapezoidal rule would finally lead to a third-order convergence.

Theorem 4.2. Let $f:D\subseteq\mathbb{R}\to\mathbb{R}$, where D is an open interval, be sufficiently differentiable function on the interval D and has a simple root $\alpha\in D$. If x_0 is sufficiently close to α, the convergence order of the method

$$x_{n+1}=x_n-\frac{2f(x_n)}{f'(x_n)+f'(u_{n+1})}, \tag{4.11}$$

in which

$$u_{n+1}=x_n-\frac{2f(x_n)}{f'(x_n)+f'(v_{n+1})} \text{ and } v_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)},$$

is three and the corresponding error equation is as

$$e_{n+1}=\frac{1}{2}c_3e_n^3+O(e_n^4),$$

where $e_n=x_n-\alpha$ and $c_j=\dfrac{f^{(j)}(\alpha)}{j!f'(\alpha)}$ for $j=1,2,3,4$.

Proof. Let α be a simple root of $f(x)$. As it is shown in the proof of Theorem 4.1, we can first obtain

$$u_{n+1}=\alpha+(\frac{1}{2}c_3+c_2^2)e_n^3+(c_4+\frac{3}{2}c_2c_3-3c_2^3)e_n^4+O(e_n^5),$$

whence

$$\begin{aligned}f'(u_{n+1})&=f'(\alpha)+\left((\frac{1}{2}c_3+c_2^2)e_n^3+(c_4+\frac{3}{2}c_2c_3-3c_2^3)e_n^4+O(e_n^5)\right)f''(\alpha)+O(e_n^6)\\&=f'(\alpha)\left(1+2((\frac{1}{2}c_3+c_2^2)e_n^3+(c_4+\frac{3}{2}c_2c_3-3c_2^3)e_n^4+O(e_n^5))\frac{f''(\alpha)}{2f'(\alpha)}+O(e_n^6)\right)\\&=f'(\alpha)\left(1+(c_2c_3+2c_2^3)e_n^3+2c_2(c_4+\frac{3}{2}c_2c_3-3c_2^3)e_n^4+O(e_n^5)\right).\end{aligned} \tag{4.12}$$

Therefore, sum of (4.2) and (4.12) yields

$$f'(x_n)+f'(u_{n+1})=f'(\alpha)\Big(2+2c_2e_n+3c_3e_n^2+(4c_4+c_2c_3+2c_2^3)e_n^3+O(e_n^4)\Big). \tag{4.13}$$

By referring to equations (4.1) and (4.13) we now obtain

$$\frac{2f(x_n)}{f'(x_n)+f'(u_{n+1})}$$
$$=\left(e_n+c_2e_n^2+c_3e_n^3+c_4e_n^4+O(e_n^5)\right)\left(1+c_2e_n+\frac{3}{2}c_3e_n^2+(2c_4+\frac{1}{2}c_2c_3+c_2^3)e_n^3+O(e_n^4)\right)^{-1}$$
$$=\left(e_n+c_2e_n^2+c_3e_n^3+c_4e_n^4+O(e_n^5)\right)\times$$
$$\Big(1-\left(c_2e_n+\frac{3}{2}c_3e_n^2+(2c_4+\frac{1}{2}c_2c_3+c_2^3)e_n^3+O(e_n^4)\right)$$
$$+\left(c_2e_n+\frac{3}{2}c_3e_n^2+(2c_4+\frac{1}{2}c_2c_3+c_2^3)e_n^3+O(e_n^4)\right)^2$$
$$-\left(c_2e_n+\frac{3}{2}c_3e_n^2+(2c_4+\frac{1}{2}c_2c_3+c_2^3)e_n^3+O(e_n^4)\right)^3+O(e_n^4)\Big)$$
$$=\left(e_n+c_2e_n^2+c_3e_n^3+c_4e_n^4+O(e_n^5)\right)\left(1-c_2e_n+(c_2^2-\frac{3}{2}c_3)e_n^2+(\frac{5}{2}c_2c_3-2c_4-c_2^3)e_n^3+O(e_n^4)\right)$$
$$=e_n-\frac{1}{2}c_3e_n^3+O(e_n^4). \qquad (4.14)$$

Finally, replacing (4.14) in relation

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

gives

$$e_{n+1}+\alpha=e_n+\alpha-\Big(e_n-\frac{1}{2}c_3e_n^3+O(e_n^4)\Big),$$

which is equivalent to

$$e_{n+1}=\frac{1}{2}c_3e_n^3+O(e_n^4),$$

and approves a third-order convergence of the method (4.11). ■

5. NUMERICAL RESULTS

In this section, we employ the proposed method (PM) increasing the convergence order of AN method [16] to solve the nonlinear equations taken from [12] and compare it with the Classical Newton's method (1.2) (CN), the Arithmetic mean Newton's method (1.4) (AN), the Harmonic mean Newton's method (1.5) (HN) and the Geometric mean Newton's method (1.6) (GN).

The results are shown in Tables 1 and 2. Displayed in Table 1 the number of iterations (n) to approximate the simple root and the number of function evaluations (NOFE), i.e. the sum of the number of function evaluations plus the number of evaluations of function derivative.

In Table 2, we have computed the computational order of convergence (COC) which is considered as ρ_n [16] if $100\,|\rho_n-\rho_{n-1}|/\min(\rho_n,\rho_{n-1})\le 10$ and ρ_1 if $n=2$, where according to (1.3),

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

The notation ND (not defined) is used when $100\,|\rho_n-\rho_{n-1}|/\min(\rho_n,\rho_{n-1})>10$. Also the multiplicity of the root is denoted by m and the stopping criterion is considered as $|x_n-\alpha|+|f(x_n)|<10^{-7}$. The results of Table 1 show that the PM needs less number of iterations to reach the desired accuracy with respect to other iterative methods. Similar to other methods in Table 2, the PM is of order one when m is greater than 1.

We have also employed the PM to nonlinear equations taken from [6] and compared it with some fourth-order methods applied in [6], which are defined as follows

- Jarratt's method (JM) [2]:

$$x_{n+1} = x_n - \left(1 - \frac{3}{2}\frac{f'(z_n) - f'(x_n)}{3f'(z_n) - f'(x_n)}\right)\frac{f(x_n)}{f'(x_n)},$$

where $z_n = x_n - 2f(x_n)/3f'(x_n)$.

- Traub-Ostrowski's method (TM) [18]:

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

- King's method (KM) [11]:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f(x_n) + \beta f(y_n)}{f(x_n) + (\beta - 2)f(y_n)}\frac{f(y_n)}{f'(x_n)}, \qquad \beta \in \mathbb{R},$$

- CM1 [6]:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{4f(x_n)^2 + 6f(x_n)f(y_n) + 3f(y_n)^2}{4f(x_n)^2 - 2f(x_n)f(y_n) - f(y_n)^2}\frac{f(y_n)}{f'(x_n)},$$

- CM2 [6]:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{2f(x_n) - f(y_n)}{2f(x_n) - 5f(y_n)}\frac{f(y_n)}{f'(x_n)},$$

where $y_n = x_n - f(x_n)/f'(x_n)$ in the above-mentioned methods.

The results are shown in Tables 3 and 4. The stopping criterion is when $|x_{n+1} - x_n| < 10^{-15}$ and $|f(x_{n+1})| < 10^{-15}$. In Table 3, it is evident that the total number of necessary iterations for the PM is less than others.
The test functions for numerical Tables 1 and 2, taken from [12], are as follows

$$f_1(x) = x^3 + 4x^2 - 10, \qquad \alpha = 1.365230013414097,$$

$$f_2(x) = xe^{x^2} - \sin^2 x + 3\cos x + 5, \qquad \alpha = -1.207647827130919,$$

$$f_3(x) = \sin^2 x - x^2 + 1, \qquad \alpha = -1.404491648215341,$$

$$f_4(x) = (x-2)^2(x+1), \qquad \alpha = -1 \vee \alpha = 2,$$

$$f_5(x) = (\sin x - \frac{\sqrt{2}}{2})^2(x+1), \qquad \alpha = \pi/4 \vee, \alpha = -1,$$

$$f_6(x) = x^2\sin(4x), \qquad \alpha = 0,$$

$$f_7(x) = (x-2)^3(x+2)^4, \qquad \alpha = 2 \vee \alpha = -2,$$

$$f_8(x) = \ln^2(x-2)(e^{x-3} - 1)\sin(\frac{\pi x}{3}), \qquad \alpha = 3.$$

The test functions for numerical Tables 3 and 4, taken from [6], are as follows

$f_1(x) = x^3 + 4x^2 - 10, \qquad \alpha = 1.365230013414097,$

$f_2(x) = x^2 - e^x - 3x + 2, \qquad \alpha = 0.257530285439861,$

$f_3(x) = xe^{x^2} - \sin^2 x + 3\cos x + 5, \qquad \alpha = -1.207647827130919,$

$f_4(x) = e^x \sin x + \ln(x^2 + 1), \qquad \alpha = 0,$

$f_5(x) = (x-1)^3 - 2, \qquad \alpha = 2.259921049894873,$

$f_6(x) = (x+2)e^x - 1, \qquad \alpha = -0.442854401002388,$

$f_7(x) = \sin^2 x - x^2 + 1, \qquad \alpha = 1.404491648215341.$

Table 1. Comparing "NOFE" and "n" of the PM with other third-order iterative methods.

F	x_0	M	n					NOFE				
			CN	AN	HN	GN	PM	CN	AN	HN	GN	PM
f_1	0.1	1	9	8	5	4	5	18	24	15	12	25
	2	1	4	3	3	3	2	8	12	12	12	10
f_2	-2	1	8	5	4	5	4	16	15	12	15	20
	-3	1	13	9	7	8	7	26	27	21	24	35
f_3	-1	1	5	3	3	3	3	10	12	12	12	15
	-3	1	5	3	3	3	2	10	9	9	9	10
f_4	-1.5	1	4	3	3	3	2	8	9	9	9	10
	1	2	23	14	11	13	12	46	42	33	39	60
	3	2	24	15	12	14	12	48	45	36	42	60
f_5	-0.8	1	4	3	2	2	2	8	9	6	6	10
	0.2	2	23	14	12	13	12	46	42	36	39	60
	1.2	2	22	14	11	12	11	42	42	33	36	55
f_6	-0.3	3	37	24	19	22	19	74	72	57	66	95
	0.4	3	37	24	19	22	19	74	72	57	66	95
f_7	1.4	3	38	25	20	22	20	76	75	60	66	100
	-3	4	57	38	30	34	30	114	114	90	102	150
f_8	2.6	4	57	38	30	34	28	114	114	90	102	140
	3.01	4	41	27	22	24	21	82	81	66	72	105
Total number			411	270	216	241	211	820	816	654	729	1055
Mean (rounded)			23	15	12	14	12	46	46	37	41	59

Table 2. Comparing "COC" of the PM with other third-order iterative methods

F	x_0	m	COC				
			CN	AN	HN	GN	PM
f_1	0.1	1	2.00	ND	ND	3.02	3.8880
	2	1	2.00	ND	ND	ND	3.7963
f_2	-2	1	2.00	3.00	3.01	3.00	3.4834
	-3	1	2.00	ND	3.01	3.00	ND

f_3	-1	1	2.00	ND	ND	3.01	ND
	-3	1	2.00	2.03	ND	ND	2.8821
f_4	-1.5	1	1.99	3.01	3.09	2.84	3.7757
	1	2	1.00	1.00	1.00	1.00	1.00
	3	2	1.00	1.00	1.00	1.00	1.00
f_5	-0.8	1	2.00	3.28	3.20	3.29	4.2496
	0.2	2	1.00	1.00	1.00	1.00	1.00
	1.2	2	1.00	1.00	1.00	1.00	1.00
f_6	-0.3	3	1.00	1.00	1.00	1.00	1.00
	0.4	3	1.00	1.00	1.00	1.00	1.00
f_7	1.4	3	1.00	1.00	1.00	1.00	1.00
	-3	4	1.00	1.00	1.00	1.00	1.00
f_8	2.6	4	1.00	1.00	1.00	1.00	1.00
	3.01	4	1.00	1.00	1.00	1.00	1.00

Table 3. Comparing “n” of the PM with other fourth-order iterative methods

F	x_0	n						
		CN	JM	TM	KM	CM1	CM2	PM
f_1	-0.3	55	46	46	49	9	44	7
	1	6	4	4	4	4	4	3
f_2	0	5	3	3	3	3	3	2
	1	5	3	3	3	3	3	3
f_3	-1	6	4	4	5	4	4	4
	-2	9	5	5	6	6	6	5
f_4	2	6	4	4	6	4	4	4
	-5	8	5	5	5	5	5	5
f_5	3	7	4	4	4	4	4	3
	4	8	5	5	5	5	4	4
f_6	2	9	5	5	6	6	4	4
	3.5	11	6	6	7	7	5	5
f_7	1	7	4	4	8	4	4	3
	2	6	4	4	4	4	4	3
Total number		148	102	102	115	68	98	55
Mean (rounded)		11	8	8	9	5	7	4

Table 4. Comparing “NOFE” of the PM with other fourth-order iterative methods

F	x_0	NOFE						
		CN	JM	TM	KM	CM1	CM2	PM
f_1	-0.3	110	138	138	147	27	132	35
	1	12	12	12	12	12	12	15
f_2	0	10	9	9	9	9	9	10
	1	10	9	9	9	9	9	15
f_3	-1	12	12	12	15	12	12	20
	-2	18	15	15	18	18	18	25
f_4	2	12	12	12	18	12	12	20
	-5	16	15	15	15	15	15	25
f_5	3	14	12	12	12	12	12	15
	4	16	15	15	15	15	12	20
f_6	2	18	15	15	18	18	12	20
	3.5	22	18	18	21	21	15	25
f_7	1	14	12	12	24	12	12	15
	2	12	12	12	12	12	12	15

Total number	296	306	306	345	204	294	275
Mean (rounded)	22	22	22	25	15	21	20

ACKNOWLEDGEMENTS

The work of the first author has been supported by the Alexander von Humboldt Foundation under the grant number: Ref 3.4 - IRN - 1128637 - GF-E.

REFERENCES

1. M. Abramowitz and I. A. Stegun, "Handbook of mathematical functions with formulas, graphs and mathematical tables", Washington: U.S. Department of Commerce. XIV, 1046 p. (**1964**); Table Errata Math. Comput. 21, 747.

2. I. K. Argyros, D.Chen and Q. Qian, "The Jarratt method in Banach space setting", *J. Comput. Appl. Math.*, **1994**, *51*, 103-106.

3. E. Babolian, M. Masjed-Jamei and M. R. Eslahchi, "On numerical improvement of Gauss-Legendre quadrature rules", *Appl. Math. Comput.*, **2005**, *160(3)*, 779-789.

4. C. Chun, "A geometric construction of iterative formulas of order three", *Appl. Math. Lett.*, **2010**, *23*,512-516.

5. C. Chun, "A simply constructed third-order modifications of Newton's method", *J. Comput. Appl. Math.*, **2008**, *219*, 81-89.

6. C. Chun and Y. Ham, "Some fourth-order modifications of Newton's method", *Appl. Math. Comput.,* **2008**, *197*,654-658.

7. L. Fang and G. Hec, "Some modifications of Newton's method with higher-order convergence for solving nonlinear equations", *J. Comput. Appl. Math.*, **2009**, *228*, 296-303.

8. M. Frontini and E. Sormani, "Some variant of Newton's method with third-order convergence", *Appl. Math. Comput.,* **2003**, *140*, 419-426.

9. H. H. H. Homeier, "A modified Newton method for root finding with cubic convergence", *J. Comput. Appl. Math*., **2003**, *157*, 227-230.

10. H. H. H. Homeier, "On Newton-type methods with cubic convergence", *J. Comput. Appl. Math.,* **2005**, *176*, 425-432.

11. R. King, "A family of fourth-order methods for nonlinear equations", *SIAM J. Numer. Anal*., **1973**, *10(5)*, 876-879.

12. T. Lukic and N. M. Ralevic, “Geometric mean Newton's method for simple and multiple roots”, *Appl. Math. Lett.*, **2008**, *21*, 30-36.

13. M. Masjed-Jamei, “New error bounds for Gauss-Legendre quadrature rules”, *Filomat*, **2014**, *28:6*, 1281-1293.

14. M. Masjed-Jamei and I. Area, “Error bounds for Gaussian quadrature rules using linear kernels”, *Int. J. Comput. Math.*, **2016**, *93*, 1505-1523.

15. M. Masjed-Jamei and G. V. Milovanovic, “Weighted Hermite quadrature rules”, *Electron. Trans. Numer. Anal.*, **2016**, *45*, 476-498.

16. A. Y. Ozban, “Some new variants of Newton's method”, *Appl. Math. Lett.*, **2004**, *17*, 677-782.

17. J. Stoer and R. Bulirsch, “An Introduction to Numerical Analysis”, Springer-Verlag, New York, **2002**.

18. J. F. Traub, “Iterative Methods for the Solution of Equations”, Chelsea publishing company, New York, **1977**.

19. R. Wait, “The Numerical Solution of Algebraic Equations”, John Wiley & Sons, **1979**.

20. S. Weerakoon and G. I. Fernando, “A variant of Newton's method with accelerated third-order convergence”, *Appl. Math. Lett.*, **2000**, *17*, 87-93.