Jordan Journal of Mathematics and Statistics (JJMS) 8(1), 2015, pp 43 - 57

NEWTON AND STEFFENSEN TYPE METHODS WITH FLEXIBLE ORDER OF CONVERGENCE

DIVYA JAIN

Abstract. New families of Newton and Steffensen type methods are derived by amalgamating known methods. The methods in the new families are of higher order than the methods amalgamated. The technique shows that it is possible to develop methods of any desired order.

1. Introduction

Non-linear equations arise in almost all areas of sciences, in particular, in physical and mathematical sciences. Most of the times, it is not possible to solve these equations analytically. Therefore, iterative methods are employed to get approximate solutions of non-linear equations. One of the standard methods is the well known Newton method

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

In the last two decades, numerous methods have been obtained by several mathematicians and these methods have been claimed to be improvement over the previously obtained methods. As examples, please refer to [2], [3], [4], [7] and the standard books [1] and [6].

1991 *Mathematics Subject Classification.* 65H05.

Key words and phrases. Newton method, secant method, Steffensen method, order of convergence, derivative free method.

Received: Nov. 15, 2014 Accepted: Feb. 3, 2015 .

In the present paper, we too contribute in this direction and obtain families of Newton and Steffensen type methods.

Very recently, Li, Peng, Zhou and Gao [5] obtained a family of family of Newton-type method given by

$$x_{n+1} = u_{n+1} - \frac{f(u_{n+1})}{f'(v_{n+1})}, \tag{1.1}$$

where $\{u_n\}$ and $\{v_n\}$ are the sequences of iterates of two methods for solving $f(x) = 0$ having order of convergence p and q respectively. It was proved that if f has a simple zero and $p > q$, then the method (1.1) is of order at least $p + q$. It was remarked in [5] that this method is useful because using this, one can generate a method of any desired order of convergence. Indeed, in order to generate a method of order r, one needs to plug in two known methods of order p and q with $p + q = r$ into (1.1).

Let us point out that while establishing the convergence of the method (1.1), the authors made an assumption that in the two methods that generate the sequences $\{u_n\}$ and $\{v_n\}$ as well as in the method (1.1) itself, the error at each corresponding iterate is the same. However, this need not be the case. As one of the aims, in the present paper, we shall remove this assumption.

2. Amalgamation of methods

In this section, we reinvestigate method (1.1) for its order of convergence. We prove the following:

Theorem 2.1. *Let f be a function having sufficient number of derivatives in a neighborhood of α which is a simple root of the equation $f(x) = 0$. Let $\{u_n\}$, $\{v_n\}$ be the sequences of iterates of two different methods for solving $f(x) = 0$ having order of convergence p, q respectively with $p \geq q$. Then the method* (1.1) *is of order $p + q$.*

Remark 2.2. Theorem 2.1 was already proved in [5]. But in their proof, the authors assumed that when $n \to \infty$

$$u_n - \alpha = v_n - \alpha = x_n - \alpha,$$

i.e., the error in the methods which generate the sequences $\{u_n\}$ and $\{v_n\}$ as well as in (1.1) is the same at the corresponding iterates. But in practice, one calculates the error only up to a finite stage and there, the three errors can be different. Our proof below takes care of this restriction.

Proof of Theorem 2.1 By the assumption

$$x_{n+1} = u_{n+1} - \frac{f(u_{n+1})}{f'(v_{n+1})}, \tag{2.1}$$

where

$$u_{n+1} = F(x_n) \tag{2.2}$$

and

$$v_{n+1} = G(x_n) \tag{2.3}$$

for some iteration functions F and G. Let a_n and b_n be the errors at the nth-iterate in the methods (2.2) and (2.3) respectively, i.e.,

$$u_n = \alpha + a_n, \quad v_n = \alpha + b_n.$$

Since the methods (2.2) and (2.3) are of order p and q respectively, the corresponding error equations can be written as

$$a_{n+1} = A_1 e_n^p + A_2 e_n^{p+1} + ... + o(e_n^{2p})$$

and

$$b_{n+1} = B_1 e_n^q + B_2 e_n^{q+1} + ... + o(e_n^{2q}),$$

where e_n is the error at the nth-iterate x_n, i.e. $e_n = x_n - \alpha$ and A_n's and B_n's are certain constants.

We have by Taylor's expansion

$$\begin{aligned} f(u_{n+1}) &= f(\alpha + a_{n+1}) \\ &= a_{n+1} f'(\alpha) + a_{n+1}^2 \frac{f''(\alpha)}{2} + o(a_{n+1}^3) \\ &= f'(\alpha)\left[a_{n+1} + C a_{n+1}^2 + o(a_{n+1}^3)\right], \end{aligned}$$

where $C = \dfrac{f''(\alpha)}{2f'(\alpha)}$. Also

$$\begin{aligned} f'(v_{n+1}) &= f'(\alpha + b_{n+1}) \\ &= f'(\alpha) + b_{n+1} f''(\alpha) + o(b_{n+1}^2) \\ &= f'(\alpha)\left[1 + 2C b_{n+1} + o(b_{n+1}^2)\right] \end{aligned}$$

so that

$$\begin{aligned} \frac{f(u_{n+1})}{f'(v_{n+1})} &= \frac{a_{n+1} + C a_{n+1}^2 + o(a_{n+1}^3)}{1 + 2C b_{n+1} + o(b_{n+1}^2)} \\ &= \left[a_{n+1} + C a_{n+1}^2 + o(a_{n+1}^3)\right]\left[1 + 2C b_{n+1} + o(b_{n+1}^2)\right]^{-1} \\ &= \left[a_{n+1} + C a_{n+1}^2 + o(a_{n+1}^3)\right]\left[1 - 2C b_{n+1} + o(b_{n+1}^2)\right] \\ &= \left[a_{n+1} - 2C a_{n+1} b_{n+1} + o(a_{n+1}^2)\right]. \end{aligned}$$

Thus the error equation for the method (1.1) can be written as

$$\begin{aligned} e_{n+1} &= a_{n+1} - \left[a_{n+1} - 2C a_{n+1} b_{n+1} + o(a_{n+1}^2)\right] \\ &= 2C a_{n+1} b_{n+1} + o(a_{n+1}^2) \\ &= 2C A_1 B_1 e_n^{p+q} + o(e_n^{p+q+1}). \end{aligned}$$

Noting that since $p \geq q$, we find that $2p \geq p + q$ and the assertion is proved. □

If we consider $u_n = v_n$ for all n in (1.1), then we immediately have the following:

Corollary 2.3. *Let f be a function having sufficient number of derivatives in a neighborhood of α which is a simple root of the equation $f(x) = 0$. Let $\{u_n\}$ be a sequence of iterates of a method for solving $f(x) = 0$ having order of convergence p. Then the method*

$$x_{n+1} = u_{n+1} - \frac{f(u_{n+1})}{f'(u_{n+1})} \tag{2.4}$$

is of order $2p$.

Next, corresponding to the method (2.4), we propose the following Steffensen type method by replacing $f'(u_{n+1})$ with the ratio $\dfrac{f(u_{n+1} + f(u_{n+1})) - f(u_{n+1})}{f(u_{n+1})}$:

$$x_{n+1} = u_{n+1} - \frac{f(u_{n+1})^2}{f(u_{n+1} + f(u_{n+1})) - f(u_{n+1})}, \tag{2.5}$$

which can be made derivative free if the method that generates the sequence $\{u_n\}$ is chosen to be derivative free. We prove the following:

Theorem 2.4. *Let f be a function having sufficient number of derivatives in a neighborhood of α which is a simple root of the equation $f(x) = 0$. Let $\{u_n\}$ be a sequence of iterates of a method for solving $f(x) = 0$ having order of convergence p. Then the method* (2.5) *is of order $2p$.*

Proof. As in the proof of Theorem 2.1, let a_n be the error at the nth iterate in the method that generates the sequence $\{u_n\}$, i.e., $u_n = \alpha + a_n$. Then

$$a_{n+1} = A_1 e_n^p + A_2 e_n^{p+1} + ... + o(e_n^{2p}), \tag{2.6}$$

where e_n is the error in the nth iterate x_n of the method (2.5). Now, by Taylor series expansion, we get

$$f(u_{n+1}) = a_{n+1} f'(\alpha) + a_{n+1}^2 \frac{f''(\alpha)}{2} + o(a_{n+1}^3)$$

so that

$$f(u_{n+1})^2 = a_{n+1}^2 f'(\alpha)^2 + o(a_{n+1}^3).$$

We also have

$$\begin{aligned}
&f(u_{n+1} + f(u_{n+1})) \\
&\quad = \big[a_{n+1}\big(1 + f'(\alpha)\big) + a_{n+1}^2 \frac{f''(\alpha)}{2} + o(a_{n+1}^3)\big] f'(\alpha) \\
&\qquad + \big[a_{n+1}\big(1 + f'(\alpha)\big) + a_{n+1}^2 \frac{f''(\alpha)}{2} + o(a_{n+1}^3)\big]^2 \frac{f''(\alpha)}{2} + o(a_{n+1}^3) \\
&\quad = a_{n+1}\big(1 + f'(\alpha)\big) f'(\alpha) + a_{n+1}^2 \Big[\frac{f''(\alpha) f'(\alpha)}{2} + \big(1 + f'(\alpha)\big)^2 \frac{f''(\alpha)}{2}\Big] + o(a_{n+1}^3).
\end{aligned}$$

Therefore

$$f(u_{n+1} + f(u_{n+1})) - f(u_{n+1}) = a_{n+1} f'(\alpha)^2 \Big[1 + C a_{n+1}\big(f'(\alpha) + 3\big) + o(a_{n+1}^2)\Big],$$

where $C = \dfrac{f''(\alpha)}{2f'(\alpha)}$. Consequently, the error equation for the method (2.5) can be written as

$$\begin{aligned}
e_{n+1} &= a_{n+1} - \frac{a_{n+1}^2 f'(\alpha)^2 + o(a_{n+1}^3)}{a_{n+1} f'(\alpha)^2 \Big[1 + C a_{n+1}\big(f'(\alpha) + 3\big) + o(a_{n+1}^2)\Big]} \\
&= a_{n+1} - \Big[a_{n+1} + o(a_{n+1}^2)\Big] \Big[1 + C a_{n+1}\big(f'(\alpha) + 3\big) + o(a_{n+1}^2)\Big]^{-1} \\
&= a_{n+1} - \Big[a_{n+1} + o(a_{n+1}^2)\Big] \Big[1 - C a_{n+1}\big(f'(\alpha) + 3\big) + o(a_{n+1}^2)\Big] \\
&= C\big(f'(\alpha) + 3\big) a_{n+1}^2 + o(a_{n+1}^3)
\end{aligned}$$

which on using (2.6) gives

$$e_{n+1} = A_1 C\big(f'(\alpha) + 3\big) e_n^{2p} + o(e_n^{2p+1}).$$

Hence, the order of the method (2.5) is at least $2p$. □

Remark 2.5. It is possible to consider a method more general than (2.5) by taking two different iterative methods. If $\{u_n\}$ and $\{v_n\}$ be sequences of iterates of two such methods, then the corresponding method looks like

$$x_{n+1} = u_{n+1} - \frac{f(u_{n+1})f(v_{n+1})}{f(v_{n+1} + f(v_{n+1})) - f(v_{n+1})}. \tag{2.7}$$

Using the arguments as in Theorems 2.1 and 2.4, we can prove the following convergence result for the method (2.7). We omit the details for conciseness.

Theorem 2.6. *Let f be a function having sufficient number of derivatives in a neighborhood of α which is a simple root of the equation $f(x) = 0$. Let $\{u_n\}$, $\{v_n\}$ be the sequence of iterates of two different methods for solving $f(x) = 0$ having order of convergence p, q respectively with $p \geq q$. Then the method* (2.7) *is of order $p + q$.*

3. Increasing the Order of Convergence

In this section, we consider methods with order of convergence higher than the ones considered in the previous section. In this direction, we consider the following method:

$$x_{n+1} = u_{n+1} - \frac{f(u_{n+1})}{f'(u_{n+1}) - \xi f(u_{n+1})} \tag{3.1}$$

for a suitable real number ξ (to be found). Note that for $\xi = 0$, (3.1) is just the method (2.4). We prove below that if the method generating the iterative sequence $\{u_n\}$ has the order p, then for all ξ, the method (3.1) is of order $2p$ while for $\xi = \dfrac{f''(u_{n+1})}{2f'(u_{n+1})}$, it is of order $3p$.

Theorem 3.1. *Let f be a function having sufficient number of derivatives in a neighborhood of α which is a simple root of the equation $f(x) = 0$. Let $\{u_n\}$ be a sequence of iterates of a method for solving $f(x) = 0$ having order of convergence p. Then, for all ξ, the method* (3.1) *is of order $2p$. Moreover, if $\xi = \dfrac{f''(u_{n+1})}{2f'(u_{n+1})}$, then the method is of order $3p$.*

Proof. Let us rewrite (3.1) as

$$x_{n+1} = u_{n+1} - \frac{f(u_{n+1})}{f'(u_{n+1})\left[1-\xi\frac{f(u_{n+1})}{f'(u_{n+1})}\right]}$$
$$= u_{n+1} - \frac{f(u_{n+1})}{f'(u_{n+1})}\left[1-\xi\frac{f(u_{n+1})}{f'(u_{n+1})}\right]^{-1}$$
$$= u_{n+1} - \frac{f(u_{n+1})}{f'(u_{n+1})}\left[1+\xi\frac{f(u_{n+1})}{f'(u_{n+1})}+\xi^2\left\{\frac{f(u_{n+1})}{f'(u_{n+1})}\right\}^2+\dots\right]$$
$$= u_{n+1} - \frac{f(u_{n+1})}{f'(u_{n+1})} - \xi\left\{\frac{f(u_{n+1})}{f'(u_{n+1})}\right\}^2 - \dots \tag{3.2}$$

As in the proof of Theorem 2.4, we can take $u_n = \alpha + a_n$ so that

$$a_{n+1} = A_1 e_n^p + A_2 e_n^{p+1} + \dots + o(e_n^{2p}). \tag{3.3}$$

By Taylor's expansion

$$f(u_{n+1}) = a_{n+1}f'(\alpha) + a_{n+1}^2\frac{f''(\alpha)}{2} + o(a_{n+1}^3)$$

and

$$f'(u_{n+1}) = f'(\alpha) + a_{n+1}f''(\alpha) + a_{n+1}^2\frac{f^{(3)}(\alpha)}{2} + o(a_{n+1}^3)$$

so that

$$\frac{f(u_{n+1})}{f'(u_{n+1})} = \left[a_{n+1} + a_{n+1}^2\frac{f''(\alpha)}{2f'(\alpha)} + o(a_{n+1}^3)\right]\left[1 + a_{n+1}\frac{f''(\alpha)}{f'(\alpha)} + o(a_{n+1}^2)\right]^{-1}$$
$$= \left[a_{n+1} + Ca_{n+1}^2 + o(a_{n+1}^3)\right]\left[1 - 2Ca_{n+1} + o(a_{n+1}^2)\right]$$
$$= a_{n+1} - Ca_{n+1}^2 + o(a_{n+1}^3), \tag{3.4}$$

where $C = \dfrac{f''(\alpha)}{2f'(\alpha)}$. Therefore

$$\left[\frac{f(u_{n+1})}{f'(u_{n+1})}\right]^2 = a_{n+1}^2 + o(a_{n+1}^3). \tag{3.5}$$

Using (3.4) and (3.5) in (3.2), we find that the error equation of the method (3.1) reads

$$e_{n+1} = (C - \xi)a_{n+1}^2 + o(a_{n+1}^3) \tag{3.6}$$

which on using (3.3) becomes

$$e_{n+1} = A_1^2(C - \xi)e_n^{2p} + o(e_n^{2p+1}).$$

This shows that the method (3.1) is of order at least $2p$.

Moreover, if $\xi = C$, i.e., $\xi = \dfrac{f''(\alpha)}{2f'(\alpha)}$, then in view of (3.6)

$$e_{n+1} \approx o(a_{n+1}^3) \approx o(e_n^{3p}),$$

i.e., the method will become of order at least $3p$ This suggests that in order to achieve $3p$ order of convergence of the method (3.1), we must take

$$\xi = \frac{f''(u_{n+1})}{2f'(u_{n+1})}$$

and the proof is complete. □

Remark 3.2. According to Theorem 3.1, we have proved that the following method is of order at least $3p$:

$$x_{n+1} = u_{n+1} - \frac{2f(u_{n+1})f'(u_{n+1})}{2f'(u_{n+1})^2 - f(u_{n+1})f''(u_{n+1})} \tag{3.7}$$

Next, as done in (2.5), we replace $f'(u_{n+1})$ by the ratio $\dfrac{f(u_{n+1} + f(u_{n+1})) - f(u_{n+1})}{f(u_{n+1})}$ in method (3.1) to obtain the following Steffensen type method:

$$x_{n+1} = u_{n+1} - \frac{f(u_{n+1})^2}{f(u_{n+1} + f(u_{n+1})) - f(u_{n+1}) - \xi f(u_{n+1})^2}. \tag{3.8}$$

On the lines similar to Theorem 2.4, we can prove the following theorem:

Theorem 3.3. *Let f be a function having sufficient number of derivatives in a neighborhood of α which is a simple root of the equation $f(x) = 0$. Let $\{u_n\}$ be a sequence of iterates of a method for solving $f(x) = 0$ having order of convergence p. Then, for all ξ, the method (3.8) is of order $2p$. Moreover, if $\xi = \dfrac{f''(u_{n+1})}{2f'(u_{n+1})}$, then the method is of order $3p$.*

4. Examples

Weerakoon and Fernando [7] obtained the following third order method:

$$x_{n+1} = x_n - \frac{2f(x_n)}{f'(x_n) + f'\left(x_n - \frac{f(x_n)}{f'(x_n)}\right)}, \tag{4.1}$$

using the trapezoidal rule and another third order method:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'\left(x_n - \frac{f(x_n)}{2f'(x_n)}\right)}, \tag{4.2}$$

using the mid-point rule. If $\{u_n\}$denote the sequence of iterates obtained from (4.2), then the method corresponding to (2.7) is given by

$$x_{n+1} = u_n - \frac{2f(u_n)}{f'(u_n) + f'\left(u_n - \frac{f(u_n)}{f'(u_n)}\right)}, \tag{4.3}$$

where

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

which is of order 6.

As a demonstration, we consider the equation

$$x^4 - x - 10 = 0.$$

On this equation, we implement Newton's method, method (4.1) and the present method (4.3) by taking the initial approximation $x_0 = 1$ and compare. The iterations are tabulated in Table 1 while the corresponding errors are presented in Table 2. One can see that the present method (4.3) converges much faster.

TABLE 1. Approximation of root for $x^4 - x - 10 = 0$

n	Newton's method	Method (4.1)	Present Method (4.3)
1.	4.333333333333333	1.0141600944238205	1.07882976880465
2.	4.0933256159208504	1.0318238873221415	1.299711307249156
3.	3.8556641054445349	1.0546924411883298	1.7334833236262615
4.	3.6210723963257019	1.0857899224096075	1.8322768727999328
5.	3.3905434513600063	1.1309437577774215	1.850565597217553
6.	3.1654491343634539	1.2023033405710548	1.8544781757117466
7.	2.9476888240388743	1.3251897883310351	1.855339454918026
8.	2.7398758224381905	1.5279704048002487	1.8555302049741234
9.	2.5455309907024075	1.7187748390408146	1.8555724842689785
10.	2.3691800726439118	1.7979624763160538	1.8555818450978918
11.	2.2161151347155212	1.8297659548859817	1.8555839540167354
12.	2.0914433355868671	1.843694619393581	1.855584423385485
13.	1.9982167161074493	1.8500420917516538	1.8555845143971677
14.	1.9353735328593944	1.8529865488370527	1.8555845423493709
15.	1.8973715686618002	1.8543635942351657	1.8555845703015723
16.	1.8764677122429578	1.8550100769967048	1.855584535194281
17.	1.8657276625888046	1.8553141104870268	1.8555845631464825
18.	1.8604353634996449	1.8554571734843375	1.8555845280391914
19.	1.8578862078054681	1.8555245416711312	1.8555845559913944
20.	1.8566724728711732	1.855556276052482	1.8555845208841064
21.	1.8560977939823635	1.8555712354546152	1.8555845488363096
22.	1.8558264633531192	1.85557824910633	1.8555845137290254
23.	1.8556985087061941	1.8555815421920465	1.8555845416812284
24.	1.8556382471551454	1.8555831326589154	1.8555845696334299
25.	1.855609823381694	1.8555838402902076	1.8555845345261386

Table 2. Errors for $x^4 - x - 10 = 0$.

n	Newton's method	Method (4.1)	Present Method (4.3)
1.	338.27166748046875	-9.9563055038452148	-9.7242279052734375
2.	266.64724731445312	-9.8983221054077148	-8.4461469650268555
3.	207.1466064453125	-9.8173122406005859	-2.7036721706390381
4.	158.30778503417969	-9.6958913803100586	-0.56122750043869019
5.	118.76252746582031	-9.4950160980224609	-0.12272926419973373
6.	87.236320495605469	-9.112736701965332	-0.027144279330968857
7.	62.548763275146484	-8.2412042617797852	-0.0060180779546499252
8.	43.613964080810547	-6.0771760940551758	-0.0013325407635420561
9.	29.441347122192383	-2.9915552139282227	-0.00029631005600094795
10.	19.136743545532227	-1.3478143215179443	-6.5050771809183061e-005
11.	11.903443336486816	-0.62037193775177002	-1.528563916508574e-005
12.	7.0416145324707031	-0.28906825184822083	-3.5761715935223037e-006
13.	3.9447953701019287	-0.13546945154666901	-6.4880327954597306e-007
14.	2.0946757793426514	-0.063657492399215698	-6.4880327954597306e-007
15.	1.0627633333206177	-0.029951976612210274	2.278565716551384e-006
16.	0.5218961238861084	-0.014100822620093822	-6.4880327954597306e-007
17.	0.25121361017227173	-0.0066384109668433666	2.278565716551384e-006
18.	0.11960681527853012	-0.0031267437152564526	-6.4880327954597306e-007
19.	0.05663052573800087	-0.0014730410184711218	-6.4880327954597306e-007
20.	0.026741910725831985	-0.00069441867526620626	-6.4880327954597306e-007
21.	0.01261004526168108	-0.00032558312523178756	-6.4880327954597306e-007
22.	0.0059431195259094238	-0.00015287118731066585	-6.4880327954597306e-007
23.	0.0027981840539723635	-7.3832838097587228e-005	-6.4880327954597306e-007
24.	0.0013196541694924235	-3.2849824492586777e-005	2.278565716551384e-006
25.	0.0006199665367603302	-1.8213004295830615e-005	-6.4880327954597306e-007

Let us consider another example. We take the equation $3x + \sin x - e^x = 0$. On this equation, again, we implement Newton's method, method (4.1) and the present method (4.3) by taking the initial approximation $x_0 = 1$ and compare. The iterations and the corresponding errors are tabulated, respectively, in Table 3 and Table 4.

TABLE 3. Approximation of root for $3x + \sin x - e^x = 0$

n	Newton's method	Method (4.1)	Present Method (4.3)
1.	-0.36637620398202769	0.44703306891956329	0.36130186641152723
2.	0.29731095663884266	0.36046612038062703	0.36042171599098644
3.	0.35913370270596362	0.36042170071144214	0.36042171738015666
4.	0.36042112392098186	0.36042171630718917	0.36042168896700311
5.	0.36042170576072896	0.36042170210061236	0.36042169035617333
6.	0.36042169155415216	0.36042171769635939	0.36042169174534355
7.	0.36042170714989918	0.36042170348978259	0.36042169313451378
8.	0.36042169294332238	0.36042168928320584	0.360421694523684
9.	0.36042170853906941	0.36042170487895286	0.36042169591285422
10.	0.3604216943324926	0.36042169067237606	0.3604216973020245
11.	0.36042170992823963	0.36042170626812309	0.36042169869119473
12.	0.36042169572166283	0.36042169206154628	0.36042170008036495
13.	0.36042171131740985	0.36042170765729331	0.36042170146953517
14.	0.36042169711083305	0.36042169345071651	0.3604217028587054
15.	0.36042171270658008	0.36042170904646353	0.36042170424787562
16.	0.36042169850000333	0.36042169483988673	0.36042170563704584
17.	0.3604217140957503	0.36042171043563376	0.36042170702621612
18.	0.36042169988917355	0.36042169622905695	0.36042170841538634
19.	0.36042171548492058	0.36042171182480398	0.36042170980455657
20.	0.36042170127834378	0.36042169761822723	0.36042171119372679

Table 4. Errors for $3x + \sin x - e^x = 0$

n	Newton's method	Method (4.1)	Present Method (4.3)
1.	-2.1506049633026123	0.2097252756357193	0.0022013199049979448
2.	-0.1613508015871048	0.00011112799256807193	3.5542214504857839e-008
3.	-0.0032238487619906664	-3.9017660213858107e-008	3.5542214504857839e-008
4.	-1.4556555925082648e-006	3.5542214504857839e-008	-3.9017660213858107e-008
5.	3.5542214504857839e-008	-3.9017660213858107e-008	-3.9017660213858107e-008
6.	-3.9017660213858107e-008	3.5542214504857839e-008	-3.9017660213858107e-008
7.	3.5542214504857839e-008	3.5542214504857839e-008	-3.9017660213858107e-008
8.	-3.9017660213858107e-008	-3.9017660213858107e-008	-3.9017660213858107e-008
9.	3.5542214504857839e-008	3.5542214504857839e-008	-3.9017660213858107e-008
10.	-3.9017660213858107e-008	-3.9017660213858107e-008	-3.9017660213858107e-008
11.	3.5542214504857839e-008	3.5542214504857839e-008	-3.9017660213858107e-008
12.	-3.9017660213858107e-008	-3.9017660213858107e-008	-3.9017660213858107e-008
13.	3.5542214504857839e-008	3.5542214504857839e-008	-3.9017660213858107e-008
14.	-3.9017660213858107e-008	-3.9017660213858107e-008	3.5542214504857839e-008
15.	3.5542214504857839e-008	3.5542214504857839e-008	3.5542214504857839e-008
16.	-3.9017660213858107e-008	-3.9017660213858107e-008	3.5542214504857839e-008
17.	3.5542214504857839e-008	3.5542214504857839e-008	3.5542214504857839e-008
18.	-3.9017660213858107e-008	-3.9017660213858107e-008	3.5542214504857839e-008
19.	3.5542214504857839e-008	3.5542214504857839e-008	3.5542214504857839e-008
20.	-3.9017660213858107e-008	-3.9017660213858107e-008	3.5542214504857839e-008

References

[1] S. D. Conte and C. de Boor, *Elementary Numerical Analysis, An Algorithmic Approach*, McGraw-Hill, 1981.

[2] D. Jain, *Families of Newton-like methods with fourth-order convergence*, Int. J. Comp. Math., 90 (2013), 1072–1082.

[3] D. Jain and B. Gupta *Two step Newton and Steffensen type methods for solving non-linear equations*, Tbil. Math. J., 5 (2012), 17–29.

[4] P. Jain, *Steffensen type methods for solving non-linear equations*, Appl. Math. Comp., 194 (2007), 527–533.

[5] Z. Li, C. Peng, T. Zhou and J. Gao, *A new Newton-type method for solving Nonlinear equations with any integer order of convergence*, J. Comp. Inf. System, 7 (2011), 2371–2378.

[6] A. Ralston, *A First Course in Numerical Analysis*, McGraw-Hill, 1965.

[7] S. Weerakoon and T.G.I. Fernando, *A variant of Newton's method with accelerated third-order convergence*, Appl. Math. Lett., 13 (2000), 87–93.

University School of Basic and Applied Sciences , Guru Gobind Singh Indraprastha University, Sector 16 C, Dwarka, Delhi-110075, India

E-mail address: divya.jain@live.in