Substituting equations (2.14) and (2.16) into the second step of equation (2.4), yields

$$x_{k+1} = y_k - \frac{f(y_k)}{\beta \sum_{k=1}^{q} \psi_i f'\big(x_k + \delta_i (y_k - x_k)\big) + \Omega f'(x_k)}$$

$$= \alpha - \left(1 - \frac{1}{(\beta + \Omega)}\right) c_2 e_k^2$$

$$+\left(-2c_2^2+c_3-\frac{2}{(\beta+\Omega)^2}\left(\Omega(-2c_2^2+c_3)+\beta\sum_{i=1}^{q}\psi_i\big(c_3+c_2^2(\delta_i-2)\big)\right)\right)e_k^3$$
$$+\Bigg(4c_2^3-7c_2c_3+3c_4+\frac{1}{\beta+\Omega}(-5c_2^3+7c_2c_3-3c_4)$$
$$-\frac{1}{(\beta+\Omega)^2}(4c_2(c_2^2-c_3)\sum_{i=1}^{q}\psi_i(\Omega-\Omega\delta_i)-\frac{1}{(\beta+\Omega)^3}\left(c_2\left(4c_2^2\left(\Omega+\sum_{i=1}^{q}\psi_i(\Omega-\delta_i)\right)^2\right.\right.$$
$$\left.\left.-(\Omega+\beta)\right)\left(3c_3\Omega+\beta\left(3c_3\sum_{i=1}^{q}\psi_i(-1+\delta_i)^2+2c_2^2\delta_i\right)\right)\right)e_k^4+O\big(e_k^5\big) \qquad (2.17)$$

For equation (2.4) to converge to α with at least convergence order $\rho=4$, the coefficients of e_k^2 and e_k^3 in equation (2.17) must vanish. Set $\beta+\Omega=1$ in equation (2.17), leads to the vanishing of the coefficient of e_k^2, while the coefficient of e_k^3 expressed in terms of β becomes:

$$\mho = -2c_2^2+c_3-2\left((-1+\beta)(2c_2^2-c_3)+\beta\sum_{i=1}^{q}\psi_i(c_3-2c_2^2\delta_i)\right)$$

Setting $\mho=0$ and applying the consistency condition in equation (2.2) and (2.3), leads to obtaining $\beta=2$. This implies that $\Omega=-1$. Hence, the general error equation of the family of the iterative methods in equation (2.4) becomes

$$e_{k+1}=\alpha+\left(c_2^3+4c_2(c_2^2-c_3)\left(1+2\sum_{i=1}^{q}\psi_i(-1+\delta_i)\right)+c_2\left(4c_2^2\left(2\sum_{i=1}^{q}\psi_i(-1+\delta_i)\right)^2\right)\right.$$
$$\left.\times\left(-3c_3+2\sum_{i=1}^{q}\psi_i(3c_3(-1+\delta_i)^2+2c_2^2\delta_i)\right)\right)e_k^4+O\big(e_k^5\big) \qquad (2.18)$$

This completes the proof of Theorem 2.1. ■

Substituting the obtained values of the parameters β and Ω into equation (2.4), yield the following generalized Two-step quadrature based family of convergence order $\rho=4$ iterative method for approximating the solution of α of $f(x)=0$ given as:

$$y_k=x_k-\frac{f(x_k)}{f'(x_k)},$$
$$Z_k=y_k-\frac{f(y_k)}{2\sum_{k=1}^{q}\psi_i f'\big(x_k+\delta_i(y_k-x_k)\big)-f'(x_k)},\ k=1,2,3,\ldots \qquad (2.19)$$

for $\delta_i\in(0,1)$, $\psi_i\in\mathbb{R}\backslash\{0\}$ and satisfying equations (2.2) and (2.3).

2.1 Some concrete form of the proposed method

In this section, some concrete forms of the iterative method developed from equation (2.19) are proposed by assigning arbitrary values to the parameters q, δ_i and ψ_1 satisfying equations (2.2) and (2.3).

For $q = 1$, $\delta_1 = \frac{1}{2}$, $\psi_1 = 1$, equation (2.19) reduces to the following iterative method.

Method 1: *Given an initial guess* x_0 *, approximate the solution* α *of* $f(x) = 0$ *by the iterative scheme:*

$$y_k = x_k - \frac{f(x_k)}{f'(x_k)}$$

$$X_{k+1} = y_k - \frac{f(y_k)}{2f'\left(\frac{x_k + y_k}{2}\right) - f'(x_k)}, \tag{2.20}$$

with error equation satisfying

$$e_{k+1} = \alpha + \left(c_2^3 - \frac{3}{2}c_2c_3\right)e_k^4 + O\left(e_k^5\right) \tag{2.21}$$

and efficiency index $EFF = 1.4142$. Method 1 is a modified form of Algorithm 2.6 in [9] that has $EFF = 1.3161$.

Upon setting $q = 2$, $\delta_1 = 0$, $\delta_2 = \frac{2}{3}$, $\psi_1 = \frac{1}{4}$, and $\psi_2 = \frac{3}{4}$, in equation (2.19), the following iterative method is proposed.

Method 2: *Given an initial guess* $\boldsymbol{x_0}$ ***,*** *approximate the solution* $\boldsymbol{\alpha}$ *of* $f(x) = 0$ *by the iterative scheme:*

$$y_k = x_k - \frac{f(x_k)}{f'(x_k)}$$

$$X_{k+1} = y_k - \frac{f(y_k)}{\frac{3}{2}f'\left(\frac{x_k + 2y_k}{3}\right) - \frac{1}{2}f'(x_k)}, k = 1,2,3, \ldots \tag{2.22}$$

with error equation obtained as

$$e_{k+1} = \alpha + (c_2^3 - c_2c_3)e_k^4 + O\left(e_k^5\right) \tag{2.23}$$

and efficiency index is $EFF = 1.4142$. Method 2 is a modified form of Algorithm 2.7 in [9] that has $EFF = 1.3161$.

For $q = 3$, $\delta_1 = 0$, $\delta_2 = \frac{1}{2}$, $\delta_2 = 1$, $\psi_1 = \frac{1}{6}$, $\psi_2 = \frac{2}{3}$, and $\psi_3 = \frac{1}{6}$, in equation (2.18), the following iterative method is proposed.

Method 3: *Given an initial guess* x_0 *, approximate the solution* α *of* $f(x) = 0$ *by the iterative scheme:*

$$y_k = x_k - \frac{f(x_k)}{f'(x_k)}$$

$$Z_k = y_k - \frac{f(y_k)}{\frac{4}{3}f'\left(\frac{x_k + y_k}{2}\right) + \frac{1}{3}f'(y_k) - \frac{2}{3}f'(x_k)}, k = 1,2,3, \ldots \quad (2.24)$$

with error equation satisfying

$$e_{k+1} = \alpha + (c_2^3 - c_2 c_3)e_k^4 + O(e_k^5) \quad (2.25)$$

and efficiency index is $EFF = 1.3195$. Method 3 is a modified form of Algorithm 2.8 in [9] that has $EFF = 1.2457$.

3. Numerical Results

In this section, the proposed iterative methods (Method 1 ($M1$), Method 2 ($M2$) and Method 3 ($M3$)), are implemented on two problems (Problem 1-2) in literature in other to illustrate their efficiency. We compare the performance of the proposed methods with the methods developed in [9] (Algorithm 2.6 (Alg. 2.6), Algorithm 2.7 (Alg. 2.7) and Algorithm 2.8 (Alg. 2.8)). Note that $M1$ is a modification of Alg. 2.6, $M2$ is a modification of Alg. 2.7 and $M3$ is a modification of Alg. 2.8. Numerical computations are performed by PYTHON 2.7.12 program with 25 digits floating arithmetic. Intel Celeron(R) CPU 1.6 GHz with 2 GB of RAM processor was used to execute all programs. The stopping criterion used for all programs is $|f(x_k)| < 10^{-15}$. For comparison, the following measurements were used:

i. number of iteration (IT)
ii. total number of functional evaluations (NFE),
iii. function of the last iteration $(|f(x_{k+1})|)$,
iv. Computer execution time in seconds (CPU).

The following problems used for implementation of the methods are taken from [2].

Problem 1

$f_1(x) = x^3 - 10, \qquad x_0 = -0.5$

Problem 2

$f_2(x) = e^{x^2+7x-30} - 1, \qquad x_0 = 3.5$

PYTHON program for Method 2 on Problem 1

```
from mpmath import*
mp.dps=25
import numpy.linalg
import time
start_time=time.clock()
x=-0.5
def f(x):
        return mpf(x**3-10)
def f_(x):
        return mpf(3*x**2)
while norm(f(x))!=0:
          y=mpf(x)-fdiv(f(x),f_(x))
          x=mpf(y)-fdiv(f(y),fdiv(3,2)*f_((x+2*y)/3)-fdiv(1,2)*f_(x))
          if norm(f(x))<=10**-15:
              break
          print((x))

print time.clock()-start_time, "seconds"
```

The program above can be altered to suit other methods in solving Problem 1 and 2. The computational results obtained by using each method are presented in Tables 3.1 and 3.2.

Table 3.1 Computational results for Problem 1

Method	**IT**	**NFE**	$\lvert f(x_{k+1})\rvert$	**CPU-time**
$Alg. 2.6$	13	52	2.9347e-07	0.045
$M1$	3	12	1.2518e-05	0.012
$Alg. 2.7$	Fail to converge			
$M2$	4	16	1.1988e-08	0.016
$Alg. 2.8$	Fail to converge			
$M3$	4	20	1.1988e-08	0.017

Table 3.2 Computational results for Problem 2

Method	**IT**	**NFE**	$\lvert f(x_{k+1})\rvert$	**CPU-time**
$Alg. 2.6$	7	28	7.3853e-14	0.028
$M1$	4	16	2.3416e-09	0.018
$Alg. 2.7$	7	28	8.2688e-13	0.030
$M2$	5	20	6.2125e-15	0.023
$Alg. 2.8$	7	35	6.7389e-13	0.032
$M3$	5	25	1.0348e-15	0.025

From Tables 3.1-3.2, the computational results indicates that the proposed iterative methods are more efficient than the methods compared. This is further supported by the fact that the CPU- times for the proposed methods are less compared with their corresponding methods in [9]. Furthermore, in Table 4.1, Alg. 2.7 and Alg. 2.8 fail to converge to the solution of Problem 1 at the given initial approximation $x_0 = -0.5$, while their corresponding proposed modified methods converged in few iterations. Thus, the proposed methods can be considered as alternative methods where the compared methods fail to converge to solution.

4. Conclusion

In this paper, a modified Algorithm 2.2 developed in Noor *et al.* [9] was proposed. The modification improves the efficiencies of all methods developed from Algorithm 2.2 in Noor *et al.* [9]. Numerical results of the proposed methods shows that they are efficient and can be used as alternative to existing methods.

Competing Interests

The authors do not have any competing interest in the manuscripts.

REFERENCES

[1] Ahmad, F., Mir, N. A., Sarikaya, M. Z., Hussain, S. and Rafiq, A. (2015). Iterative method for solving nonlinear equations using decomposition method. *J. Appl. Environ. Biol. Sci. 5(4)247-253.*

[2] Shah, A. F. and Noor, A. M. (2015). Some numerical methods for solving nonlinear equations by using decomposition technique. *Applied Mathematics and Computation; 251:378-380.*

[3] Frontini, M. and Sormani, E. (2004). Third order methods from quadrature formulae for solving systems of nonlinear equations.*Appl. Math.Comput., 149:771-782.*

[4] Noor, M.A., Noor, K.I.. and Waseem, M. (2013).Decomposition method for solving system of nonlinear equations. *Engineering Math.Lett., 2:34-41*

[5] Saqib, M. and Iqbal, M. (2017). Some multi-step iterative methods for solving nonlinear equations. *Open J. Math. Sci.* 1(1)25-33.

[6] Ogbereyivwe, O. and Muka, K. (2017). Third and Fourth order Iterative schemes for nonlinear systems using decomposition approach. *The Journal of the Mathematical Association of Nigeria, ABACUS, 44,131-147.*

[7] Ogbereyivwe, O. and Emunefe, O. J. (2014), New Three-Steps Iterative Methods for Solving Nonlinear Equations, *IOSR Journal of Mathematics, 10(5), 44-47.*

[8] Ogbereyivwe, O. and Atoma O. J. (2014), New Two-Step Method with Fifth-Order Convergence for Solving Nonlinear Equations, *International Journal of Mathematics and Statistics Invention, 2 (11), 24-27.*

[9] Noor M. A., Waseem M. and Noor K. I. (2015). New iterative technique for solving nonlinear equations. *Applied Mathematics and Computation, 265,115- 1125.*

[10] Cordero A. and Torregrosa J. R. (2007). Variants of Newton's method using fifth- order quadrature formulas. *Appl. Math.Comput. 190, 686-698.*

[11] Khirallah M. Q. and Hafiz M. A. (2012). Novel three order methods for solving a system of nonlinear equations. *Bulletin of Mathematical Sciences and Applications, 1(2)1-14.*

[12] Darvishi M. T., and Barati A. (2007). A fourth-order method from quadrature formulae to solve systems of nonlinear equations. *Appl. Math. and Comput. 188, 257-261.*

[13] Liu; Z. (2015). A new cubic convergence method for solving systems of nonlinear equations. *Intl. Journa of Applied Science and Mathematics, 2(5), 2394-2894.*

[14] Liu; Z. and Fang; Q. (2015). A new Newton-type method with third-order for solving systems of nonlinear equations. *Journal of Applied Mathematics and Physics, 3, 1256-1261.*