A new two-step optimal approach for solution of real-world models and their dynamics

Sanaullah Jamali[1*], Zubair Ahmed Kalhoro[1], Abdul Wasim Shaikh[1], Muhammad Saleem Chandio[1], Owais Ali Rajput[1], Umair Khalid Qureshi[2]

[1] Institute of Mathematics and Computer Science, University of Sindh, Jamshoro, Sindh, Pakistan
[2] Department of Business Administration, Shaheed Benazir Bhutto University, Sanghar, Sindh, Pakistan

Abstract- *In this study, a numerical method for determining the root of non-linear algebraic and transcendental equations is developed. Here, a two-step derivative free method for solving non-linear algebraic and transcendental equations is provided. This approach has an order four accuracy with three functional evaluations and no derivative. The generated scheme has an efficiency index of 1.587. The novelty is the use of Traub's method as a first and second step. Numerical examples, real-world problems, and a study of dynamics are utilized to demonstrate the performance of the provided scheme and to compare it to other approaches of the same order that are accessible in the literature. For numerical results and basin of attraction, the software MATLAB, Mathematica, and Maple are used.*

Keywords: *Non-linear Equation, Convergence Analysis, Efficiency Index, Basin of attraction, Derivative free*

I. INTRODUCTION

Finding the zeros of non-linear functions rapidly and accurately is a fundamental task in scientific computation. In general, this is the problem of solving a non-linear equation $f(x) = 0$, where $f: D \subseteq \mathbb{R} \longrightarrow \mathbb{R}$. There are certain analytical methods to solve non-linear algebraic equations $f(x) = 0$, upto 4th degree. There is no any analytical method is derived for the solution of non-linear algebraic & transcendental equations of order greater than or equal to 5. As a result, the only option to get sufficient numerical solution is through numerical methods based on iterative procedures. One of the most well-known and renowned iterative approaches for finding solution to non-linear equation is Newton's method [1].

$$x_{n+1} = \frac{f(x_n)}{f'(x_n)} \tag{1.1}$$

However the convergence of Newton's method is quadratic and there are two function evaluation required i.e. $f(x) \& f'(x)$, if $f'(x) = 0$ than this method fails, and the computational cost of methods involving derivatives are increased as compared to derivative free method that's way now the researchers are preferred derivative free methods. Steffensen developed a derivative-free iterative method [2,3].

$$w^n = x^n + f(x^n), \qquad x^{n+1} = x^n - \frac{f(x^n)}{f[x^n, w^n]}, \tag{1.2}$$

Where $[x^n, w^n] = \frac{f(x^n) - f(w^n)}{x^n - w^n}$, it maintains the Newton's method's convergence order and efficiency index.

The primary goal of this research is to develop efficient derivative-free techniques for solving non-linear equation and obtained an optimal iterative method that will support the conjecture See in [4].

A. Cordero et al. [5] presented general procedure for obtaining optimal derivative free iterative methods for nonlinear equations f(x) = 0 using polynomial interpolation. [6] Presented a new bracketing and derivative free method of quadratic convergence, derived from the newton backward interpolation technique. For solving non-linear equations, a procedure for design of Steffensen-type algorithms of various orders is proposed by A. Cordero & J. Torregrosa [7], many iterative techniques may be transformed into derivative-free iterative systems by using particular divided difference of first order has been suggested. S. Jamali [8] presented an alternate second-order bracketing and derivative free method for the solution of non-linear equations by using Stirling interpolation technique. For the solution of non-linear equation, B. Neta [9] proposed a higher order derivative-free method and the novelty in the method is using Traub's method as first step. [10] Build two families of derivative free two-point iterative algorithms for solving nonlinear equations. These methods make use of an appropriate parametric function and an arbitrary real parameter. The first family has a convergence order of four, using just three function evaluations per iteration.

[11], [12] Proposed the three step derivative free methods of order eight for solution of nonlinear algebraic and transcendental equation both methods four function evaluations with no derivative, hence both methods are optimal method. Recently, [13], [14] presented two step, three step derivative free optimal methods for solution of nonlinear algebraic and transcendental equations respectively.

II. DERIVATION OF PROPOSED METHOD

A. K. Maheshwari [15] Proposed a two-step fourth order optimal method with three function evaluation (two function and one first derivative).

i.e

$$\left.\begin{aligned}&\text{Step 1. } y_n = x_n - \frac{f(x_n)}{f'(x_n)}\\&\text{Step 2. } x_{n+1} = x_n + \left(\frac{f^2(x_n)}{f(y_n)-f(x_n)} - \frac{f^2(y_n)}{f(x_n)}\right)\frac{1}{f'(x_n)}\end{aligned}\right\} \quad (2.1)$$

Where $f'(x_n)$ is defined as (A. Cordero & J. Torregrosa [7]) as follow,

$f'(x_n) \approx f[k_n, x_n]$,

where $k_n = x_n + \gamma f^n(x_n)\ \forall\ \gamma \neq 0 \wedge n \geq 2$ (2.2)

Taking $\gamma = 1 \wedge n = 2$

$$f'(x_n) \approx f[k_n, x_n] = \frac{f(k_n)-f(x_n)}{f^2(x_n)} \quad (3)$$

Finally, we got

$$\left.\begin{aligned}&\text{Step 1. } y_n = x_n - \frac{f(x_n)}{f[k_n,x_n]}\\&\text{Step 2. } x_{n+1} = x_n + \left(\frac{f^2(x_n)}{f(y_n)-f(x_n)} - \frac{f^2(y_n)}{f(x_n)}\right)\frac{1}{f[k_n,x_n]}\end{aligned}\right\} \quad (4)$$

Equation (4) is optimal fourth order derivative free method.

III. CONVERGENCE ANALYSIS

Theorem I: Let $\alpha \in D$ be a simple zero of a sufficiently differentiable function $f : D \subset R \to R$ in an open interval D, which contains x_0 as an initial approximation of α. Then the method (4) is of order fourth and includes only three function evaluations per full iteration, and no derivatives used.

Proof.

The Taylor's series expansion of the function $f(x_n)$ can be written as:

$$f(x_n) = \sum_{m=0}^{\infty} \frac{f^m(\alpha)}{m!}(x_n-\alpha)^m$$

$$= f(\alpha) + f'(\alpha)(x_n-\alpha) + \frac{f''(\alpha)}{2!}(x_n-\alpha)^2 + \frac{f'''(\alpha)}{3!}(x_n-\alpha)^3 + \cdots \quad (5)$$

For simplicity, we assume that $A_k = \left(\frac{1}{k!}\right)\frac{f^k(\alpha)}{f'(\alpha)}, k \geq 2.$

and assume that $e_n = x_n - \alpha$. Thus, we have

$$f(x_n) = f'(\alpha)[e_n + A_2 e_n{}^2 + A_3 e_n^3 + A_4 e_n^4 + \cdots] \quad (6)$$

Furthermore, we have

$$f[k_n,\ x_n] = \frac{f(k_n)-f(x_n)}{k_n - x_n} = f'(\alpha)\big(1 + 2A_2 e_n + e_n^2(A_2 f'^2(\alpha) + 3A_3) + e_n^3\big(2A_2^2 f'^2(\alpha) + 3A_3 f'^2(\alpha) + 4A_4\big) + e_n^4\big(A_2^3 f'^2(\alpha) + 8A_2A_3 f'^2(\alpha) + A_3 f'^4(\alpha) + 6A_4 f'^2(\alpha) + 5A_5\big) + O(e_n^5)\big) \quad (7)$$

Dividing equation (6) by equation (7) gives us

$$\frac{f(x_n)}{f[k_n,x_n]} = e_n - A_2 e_n^2 + e_n^3\big(2A_2^2 - A_2 f'^2(\alpha) - 2A_3\big) + e_n^4\big(-4A_2^3 + A_2^2 f'^2(\alpha) + 7A_2A_3 - 3A_3 f'^2(\alpha) - 3A_4\big) + O(e_n^5) \quad (8)$$

And hence, we have

Step 1. $y_n = x_n - \frac{f(x_n)}{f[k_n,x_n]} = A_2 e_n^2 + e_n^3(-2A_2^2 + A_2 f'^2(\alpha) + 2A_3) + e_n^4(4A_2^3 - A_2^2 f'^2(\alpha) - 7A_2A_3 + 3A_3 f'^2(\alpha) + 3A_4) + O(e_n^5)$ (9)

Again expanding $f(y_n)$ about α, we have

$f(y_n) = f'(\alpha)[y_n + A_2 y_n{}^2 + A_3 y_n^3 + A_4 y_n^4 + O(y_n^5)]$10)

And then from equation (10), we have

$$f(y_n) = f'(\alpha)\big[A_2 e_n^2 + e_n^3(-2A_2^2 + A_2 f'^2(\alpha) + 2A_3) + e_n^4(4A_2^3 - A_2^2 f'^2(\alpha) - 7A_2A_3 + 3A_3 f'^2(\alpha) + 3A_4) + \cdots\big] \quad (11)$$

From equation (6) and equation (11) we got

$$\frac{f(x_n)}{f(y_n)-f(x_n)} = -1 - A_2 e_n + e_n^2(2A_2^2 - A_2 f'^2(\alpha) - 2A_3) - 3e_n^3(A_2^3 - 2A_2A_3 + A_3 f'^2(\alpha) + A_4) + e_n^4(3A_2^4 - 11A_2^2A_3 + 8A_2A_4 + 4A_3^2 - A_3 f'^4(\alpha) - 6A_4 f'^2(\alpha) - 4A_5) + O(e_n^5), \quad (12)$$

$$\left(\frac{f(y_n)}{f(x_n)}\right)^2 = A_2^2 e_n^2 - 2e_n^3(A_2(3A_2^2 - A_2 f'^2(\alpha) - 2A_3)) + e_n^4(25A_2^4 - 10A_2^3 f'^2(\alpha) - 32A_2^2A_3 + A_2^2 f'^4(\alpha) + 10A_2A_3 f'^2(\alpha) + 6A_2A_4 + 4A_3^2 + O(e_n^5), \quad (13)$$

$$\frac{f(x_n)}{f(y_n)-f(x_n)} - \left(\frac{f(y_n)}{f(x_n)}\right)^2 = -1 - A_2 e_n + e_n^2\big(A_2^2 - A_2 f'^2(\alpha) - 2A_3\big) + e_n^3\big(3A_2^3 - 2A_2^2 f'^2(\alpha) + 2A_2A_3 - 3A_3 f'^2(\alpha) - 3A_4\big) + e_n^4\big(-22A_2^4 + 10A_2^3 f'^2(\alpha) + 21A_2^2A_3 - A_2^2 f'^4(\alpha) - 10A_2A_3 f'^2(\alpha) + 2A_2A_4 - A_3 f'^4(\alpha) - 6A_4 f'^2(\alpha) - 4A_5\big) + {+O(e_n^5)}, \quad (14)$$

And from equation equation (12), (13) and (14)

$$\left(\frac{f(x_n)}{f(y_n)-f(x_n)} - \left(\frac{f(y_n)}{f(x_n)}\right)^2\right)\left(\frac{f(x_n)}{f[k_n,x_n]}\right) = -e_n + e_n^4\big(4A_2^3 - A_2^2 f'^2(\alpha) - A_2A_3\big) + O(e_n^5) \quad (15)$$

And finally

Step 2. $x_{n+1} = x_n + \left(\frac{f(x_n)}{f(y_n)-f(x_n)} - \left(\frac{f(y_n)}{f(x_n)}\right)^2\right)\left(\frac{f(x_n)}{f[k_n,x_n]}\right)$

$= e_n^4(4A_2^3 - A_2^2 f'^2(\alpha) - A_2A_3) + O(e_n^5)$ (16)

IV. NUMERICAL EXPERIMENT

Following problems are taken from literature[1–18] and tested in proposed methods.

Problem-1

Functions	Exact Root
$f_1(x) = \log(x^2 + x + 2) - x + 1$	4.152590736757158 …
$f_2(x) = \sin^2(x) - x + 1$	1.897194306338642 …
$f_3(x) = \tanh(x) + 2x$	0
$f_4(x) = 1 + e^{x^2+x} - \cos(-x^2 + 1) + x^3$	-1
$f_5(x) = x^3 + 2x + \sin(x)$	0
$f_6(x) = x^6 + (1-2x)^5$	1

Table 1. Shows the value of $|x_1 - x_0|, |x_2 - x_1|, |x_3 - x_2|$& COC of different methods of order Four.

$(f_n(x), x_0)$	Proposed 4th	ZLM 4th	KTM 4th	AKKB 4th	JLM 4th
$(f_1(x), 3.2)$					
$\lvert x_1 - x_0\rvert$	$9.5248e-1$	$9.5275e-1$	$9.5282e-1$	$9.5279e-1$	$9.5326e-1$
$\lvert x_2 - x_1\rvert$	$1.1437e-4$	$1.5728e-4$	$2.2959e-4$	$1.9549e-4$	$6.6543e-4$
$\lvert x_3 - x_2\rvert$	$7.5425e-21$	$5.9779e-20$	$3.6803e-19$	$1.7325e-19$	$1.2106e-16$
COC	4.1272	4.0769	4.0892	4.0816	4.0367
$(f_2(x), 2)$					
$\lvert x_1 - x_0\rvert$	$1.0280e-1$	$1.0279e-1$	$1.0279e-1$	$1.0278e-1$	$1.0279e-1$
$\lvert x_2 - x_1\rvert$	$4.4155e-6$	$1.2286e-5$	$1.8065e-5$	$2.1421e-5$	$1.8524e-5$
$\lvert x_3 - x_2\rvert$	$9.0230e-24$	$2.0611e-21$	$1.4380e-20$	$3.4123e-20$	$2.8873e-20$
COC	4.0507	4.0217	4.0210	4.0200	3.9547
$(f_3(x), 3)$					
$\lvert x_1 - x_0\rvert$	3.0465 …	2.9279 …	2.9342 …	2.9171 …	2.8963 …
$\lvert x_2 - x_1\rvert$	$4.6499e-2$	$7.2073e-2$	$6.5753e-2$	$8.2863e-2$	$1.0373e-1$
$\lvert x_3 - x_2\rvert$	$1.9509e-10$	$1.8171e-6$	$1.1727e-6$	$3.3933e-6$	$5.5389e-7$
COC	4.6121	2.8583	2.8787	2.8370	3.6465
$(f_4(x), -0.9)$					
$\lvert x_1 - x_0\rvert$	$9.9998e-2$	$1.0010e-1$	$1.0010e-1$	$1.0009e-1$	$9.9996e-2$
$\lvert x_2 - x_1\rvert$	$1.8321e-6$	$9.6052e-5$	$9.7543e-5$	$9.3232e-5$	$4.4735e-6$
$\lvert x_3 - x_2\rvert$	$9.3896e-25$	$2.1990e-16$	$2.4668e-16$	$1.7382e-16$	$1.1472e-22$
COC	3.8611	3.8571	3.8513	3.8700	3.8146
$(f_5(x), -0.2)$					
$\lvert x_1 - x_0\rvert$	$2.0000e-1$	$2.0131e-1$	$2.0097e-1$	$1.9708e-1$	$2.0029e-1$
$\lvert x_2 - x_1\rvert$	$3.6738e-6$	$1.3104e-3$	$9.7323e-4$	$2.9195e-3$	$2.9316e-4$
$\lvert x_3 - x_2\rvert$	$1.0329e-28$	$2.3849e-14$	$5.3896e-15$	$1.3095-12$	$3.3414e-19$
COC	4.2617	4.1120	4.0627	4.0750	4.2608
$(f_6(x), 1.01)$					
$\lvert x_1 - x_0\rvert$	$1.0003e-2$	$1.0010e-2$	$1.0026e-2$	$1.0096e-2$	NC
$\lvert x_2 - x_1\rvert$	$2.91653-6$	$1.0186e-5$	$2.5518e-5$	$9.6382e-5$	NC
$\lvert x_3 - x_2\rvert$	$1.8653e-20$	$1.4573e-17$	$1.5071e-15$	$8.7623e-13$	NC
COC	4.0150	3.9581	3.9428	3.9805	NC

Problem-2. Application Problems

Problem-2.1 Planck's radiation law. See In [10], [17–18]

$$exp(-x) - 1 + \frac{x}{5} = 0.$$

Initial guess $x_0 = 4m$.

Table 2. Numerical results for problem 2.1 for first four iterations and their absolute function values at $x_0 = 4$.

Method	*1st iteration*	*2nd iteration*	*3rd iteration*	*4th iteration*
Proposed 4th	4.96552765×10^{0} $7.97993466 \times 10^{-5}$	4.96511423×10^{0} $6.78093521 \times 10^{-19}$	4.96511423×10^{0} $3.53748443 \times 10^{-75}$	4.96511423×10^{0} $2.62007853 \times 10^{-300}$
ZLM 4th	4.96569540×10^{0} $1.12179790 \times 10^{-4}$	4.96511423×10^{0} $3.59465370 \times 10^{-18}$	4.96511423×10^{0} $3.79305221 \times 10^{-72}$	4.96511423×10^{0} $4.70235996 \times 10^{-288}$
KTM 4th	4.96577301×10^{0} $1.27160649 \times 10^{-4}$	4.96511423×10^{0} $6.23920096 \times 10^{-18}$	4.96511423×10^{0} $3.61957423 \times 10^{-71}$	4.96511423×10^{0} $4.09988374 \times 10^{-284}$
AKKB 4th	4.96568204×10^{0} $1.09601540 \times 10^{-4}$	4.96511423×10^{0} $3.24296845 \times 10^{-18}$	4.96511423×10^{0} $2.48770282 \times 10^{-72}$	4.96511423×10^{0} $8.61433186 \times 10^{-289}$

JLM 4^{th}	4.96548086×10^{0} $7.07684789 \times 10^{-5}$	4.96511423×10^{0} $4.00140371 \times 10^{-18}$	4.96511423×10^{0} $4.09178261 \times 10^{-74}$	4.96511423×10^{0} $4.47417822 \times 10^{-290}$

Table 3. Numerical results for problem 2.1, error fixed at $\delta = 10^{-300}$.

Method	IG	N	FE	$CPU\ Time$
Proposed 4^{th}	4	4	12	4.7×10^{-2}
ZLM 4^{th}	4	5	15	1.56×10^{-1}
KTM 4^{th}	4	5	15	1.41×10^{-1}
AKKB 4^{th}	4	5	15	1.56×10^{-1}
JLM 4^{th}	4	5	15	1.41×10^{-1}

Problem 4.2 See in [21] An engineer has estimated the depth to be x_0.

$$f(x) = \frac{x^3 + 2.87x^2 - 10.28}{4.62} - x$$

Table 4. Numerical results for problem 4.2 for first four iterations and their absolute function values at $x_0 = 2.5$.

Method	$1st\ iteration$	$2nd\ iteration$	$3rd\ iteration$	$4th\ iteration$
Proposed 4^{th}	2.02027249×10^{0} $7.48902372 \times 10^{-2}$	2.00211834×10^{0} $1.81518577 \times 10^{-6}$	2.00211878×10^{0} $5.23750052 \times 10^{-25}$	2.00211878×10^{0} $3.63026575 \times 10^{-99}$
ZLM 4^{th}	2.03243415×10^{0} $1.25773423 \times 10^{-1}$	2.00212034×10^{0} $6.37273448 \times 10^{-6}$	2.00211878×10^{0} $4.91917210 \times 10^{-23}$	2.00211878×10^{0} $1.74646684 \times 10^{-91}$
KTM 4^{th}	2.04739697×10^{0} $1.89164417 \times 10^{-1}$	2.00213489×10^{0} $6.58809383 \times 10^{-5}$	2.00211878×10^{0} $1.30087342 \times 10^{-18}$	2.00211878×10^{0} $1.97780000 \times 10^{-73}$
AKKB 4^{th}	NC NC	NC NC	NC NC	NC NC
JLM 4^{th}	2.07471783×10^{0} $3.07167637 \times 10^{-1}$	2.00212259×10^{0} $1.55938005 \times 10^{-5}$	2.00211878×10^{0} $6.80601258 \times 10^{-23}$	2.00211878×10^{0} $2.46979666 \times 10^{-92}$

Table 5. Numerical results for problem 2.2, error fixed at $\delta = 10^{-300}$.

Method	IG	N	FE	$CPU\ Time$
Proposed 4^{th}	2.5	5	15	6.30×10^{-2}
ZLM 4^{th}	2.5	5	15	1.25×10^{-1}
KTM 4^{th}	2.5	6	18	9.40×10^{-2}
AKKB 4^{th}	2.5	NC	NC	NC
JLM 4^{th}	2.5	5	15	1.09×10^{-1}

Problem-4.3 See in [21]

$$f(x) = \frac{x + \cos(x)\sin(x)}{\pi} - \frac{1}{4}.$$

Table 6. Numerical results for problem 2.3 for first four iterations and their absolute function values at $x_0 = 6$.

Method	*1st iteration*	*2nd iteration*	*3rd iteration*	*4th iteration*
Proposed 4th	$2.54009846 \times 10^{-1}$ $9.17255527 \times 10^{-2}$	$4.15707941 \times 10^{-1}$ $7.86654281 \times 10^{-5}$	$4.15855597 \times 10^{-1}$ $1.31147575 \times 10^{-16}$	$4.15855597 \times 10^{-1}$ $1.01422941 \times 10^{-63}$
ZLM 4th	1.12721679×10^{0} $2.32193972 \times 10^{-1}$	$-7.45582108 \times 10^{-1}$ $6.45976743 \times 10^{-1}$	1.76135416×10^{0} $2.51457755 \times 10^{-1}$	$-4.18211577 \times 10^{-1}$ $5.01253787 \times 10^{-1}$
KTM 4th	1.17928690×10^{0} $2.37650134 \times 10^{-1}$	$-3.11725259 \times 10^{0}$ 1.23450769×10^{0}	1.16287965×10^{0} $2.36068144 \times 10^{-1}$	$-2.57075469 \times 10^{0}$ $9.23571937 \times 10^{-1}$
AKKB 4th	$8.92979567 \times 10^{-1}$ $1.89729317 \times 10^{-1}$	$3.73875587 \times 10^{-1}$ $2.27676807 \times 10^{-2}$	$4.15855033 \times 10^{-1}$ $3.00168777 \times 10^{-7}$	$4.15855597 \times 10^{-1}$ $1.09873247 \times 10^{-27}$
JLM 4th	1.09720389×10^{0} $2.28448813 \times 10^{-1}$	$-4.28201728 \times 10^{-1}$ $5.06540994 \times 10^{-1}$	$5.90941887 \times 10^{-1}$ $8.53722068 \times 10^{-2}$	$4.15295251 \times 10^{-1}$ $2.98585238 \times 10^{-4}$

Table 7. Numerical results for problem 4.3, error fixed at $\delta = 10^{-300}$.

Method	*IG*	*N*	*FE*	*CPU Time*
Proposed 4th	6	5	15	2.50×10^{-1}
ZLM 4th	6	10	30	1.157
KTM 4th	6	12	36	1.281
AKKB 4th	6	6	18	6.41×10^{-1}
JLM 4th	6	8	24	9.84×10^{-1}

Problem 4.4 Study of the multipactor effect [19], [21].

$$\boldsymbol{f}(\boldsymbol{x}) = x - \frac{1}{2}cos(x) + \frac{\pi}{4}$$

The required zero of the above function is $x^* \approx -0.3094661392082146514$ ...

Table 8. Numerical results for problem 2.4 for first four iterations and their absolute function values at $x_0 = -2$.

Method	*1st iteration*	*2nd iteration*	*3rd iteration*	*4th iteration*
Proposed 4th	$-3.23939295 \times 10^{-1}$ $1.25355748 \times 10^{-2}$	$-3.09093270 \times 10^{-1}$ $9.66889837 \times 10^{-10}$	$-3.09093272 \times 10^{-1}$ $3.37180759 \times 10^{-38}$	$-3.09093272 \times 10^{-1}$ $4.98661610 \times 10^{-152}$
ZLM 4th	$-1.17178656 \times 10^{-1}$ $1.71648291 \times 10^{-1}$	$-3.09023885 \times 10^{-1}$ $5.88340175 \times 10^{-5}$	$-3.09093272 \times 10^{-1}$ $9.23557175 \times 10^{-19}$	$-3.09093272 \times 10^{-1}$ $5.60808220 \times 10^{-74}$
KTM 4th	$-1.36765449 \times 10^{-1}$	$-3.08996836 \times 10^{-1}$	$-3.09093272 \times 10^{-1}$	$-3.09093272 \times 10^{-1}$

	$1.53301627 \times 10^{-1}$	$8.17699775 \times 10^{-5}$	$8.99389858 \times 10^{-18}$	$1.31652868 \times 10^{-69}$
AKKB 4th	$-9.77061693 \times 10^{-2}$ $1.90076720 \times 10^{-1}$	$-3.09111931 \times 10^{-1}$ $1.58217363 \times 10^{-5}$	$-3.09093272 \times 10^{-1}$ $1.76471131 \times 10^{-21}$	$-3.09093272 \times 10^{-1}$ $2.73103535 \times 10^{-85}$
JLM 4th	$-1.11995555 \times 10^{-1}$ $1.76535083 \times 10^{-1}$	$-3.09070448 \times 10^{-1}$ $1.93518788 \times 10^{-5}$	$-3.09093272 \times 10^{-1}$ $3.16582107 \times 10^{-21}$	$-3.09093272 \times 10^{-1}$ $2.26749626 \times 10^{-84}$

Table 9. Numerical results for problem 2.4, error fixed at $\delta = 10^{-300}$.

Method	IG	N	FE	$CPU\ Time$
Proposed 4th	-2	5	15	2.66×10^{-1}
ZLM 4th	-2	6	18	2.97×10^{-1}
KTM 4th	-2	6	18	3.59×10^{-1}
AKKB 4th	-2	5	15	5.45×10^{-1}
JLM 4th	-2	5	15	2.97×10^{-1}

V. Dynamics Study of the Methods

For investigation of stability of proposed method at various initial guess we use the dynamical system i.e basin of attraction. If an algorithm fails to converge or converges to a different solution, it is considered inferior to the others. The main difficulty with this form of comparison is that the starting point is just one among an infinite number of possibilities. To combat this, the concept of a basin of attraction was developed. If a function contains n different zeroes (roots), the plane is split into n basins in an ideal case, and every basin has different color. Stewart [22] was the initially discuss the basin of attraction method. He contrasted Newton's approach to Halley's, Popovski's, and Laguerre's third order methods. This is preferable to comparing method by executing various non-linear functions with a certain initial value. Many articles have been published in the recent decade that use the concept of basin of attraction to compare the efficacy of various techniques.

VI. Basin of Attraction for Proposed Algorithms

All basins are plotted with MATLAB R2018b within the range $\Re = [-1 \times 1] \times [-1 \times 1]$ with a density of $300 \times 300 = 90{,}000$ points. To terminate iterations, an error threshold of 1×10^{-10} or a maximum count of 100 iterations is chosen. Each point in $\Re$ is then picked as the starting condition for the algorithms. If the sequence generated by the iterative algorithm converges to a root x_k^* to the function $P_i(x)$ with the specified tolerance and iterations count $N \leq 100$, we decide to give the starting point a distinct color (not black) depending on the root it converged to. If the iterative algorithm starting with $x \in \Re$ transcends 100 iterations count before converging to any root x_k or converges to some other value, say p, with specified tolerance $|p - x^*| < 1 \times 10^{-10}$, we conclude that the starting point has diverged and a black color is assigned to it.

The number of iterations is depicted for each point in another basin with reference of a color bar alongside.

S. No	**Functions** $(P(x))$	**Roots** $(x_k : k = 1, 2, 3, \dots)$
1.	$P_1(x) = x^2 - \frac{1}{4}$	$x_k = \frac{1}{2}, -\frac{1}{2}$
2.	$P_2(x) = 3x^3 - 2x$	$x_k = 0, \sqrt{\frac{3}{2}}, -\sqrt{\frac{3}{2}}$
3.	$P_3(x) = x^3 - \frac{1}{2}x^2 + \frac{1}{4}x - \frac{1}{8}$	$x_k = \frac{1}{2}, \frac{1}{2}i, -\frac{1}{2}i$
4.	$P_4(x) = x^3 + \frac{1}{16}x - \frac{5}{32}$	$x_k = \frac{1}{2}, -\frac{1}{4} \pm \frac{1}{2}i$

5.	$P_5(x) = x^4 + \frac{1}{64}$	$x_k = \frac{1 \pm 1i}{4}, \frac{-1 \pm 1i}{4} P_1(x) = x^2 - 1$
6.	$P_6(x) = x^5 - \frac{1}{2}ix^4 + \frac{1}{64}x - \frac{1}{128}i$	$x_k = \frac{1 \pm 1i}{4}, \frac{-1 \pm 1i}{4}, \frac{1}{2}i$
7.	$P_7(x) = e^x \cos x$	$x_k \cong -4.712388980384690,$ ± 1.570796326794897
8.	$P_8(x) = x^2 - 1$	$x_k = 1, -1$

Basin of attraction of proposed method:

Figure 1. Basin of attraction, left figure shows roots while right figure shows number of iterations at each initial point of $P_1(x)$ obtained by the proposed fourth order method.

Figure 2. Basin of attraction, left figure shows roots while right figure shows number of iterations at each initial point of $P_2(x)$ obtained by the proposed fourth order method.

Figure 3. Basin of attraction, left figure shows roots while right figure shows number of iterations at each initial point of $P_3(x)$ obtained by the proposed fourth order method.

Figure 4. Basin of attraction, left figure shows roots while right figure shows number of iterations at each initial point of $P_4(x)$ obtained by the proposed fourth order method.

Figure 5. Basin of attraction, left figure shows roots while right figure shows number of iterations at each initial point of $P_5(x)$ obtained by the proposed fourth order method.

Figure 6. Basin of attraction, left figure shows roots while right figure shows number of iterations at each initial point of $P_6(x)$ obtained by the proposed fourth order method.

Figure 7. Basin of attraction, left figure shows roots while right figure shows number of iterations at each initial point of $P_7(x)$ obtained by the proposed fourth order method.

Figure 8. Basin of attraction, left figure shows roots while right figure shows number of iterations at each initial point of $P_8(x)$ obtained by the proposed fourth order method.

VII. CONCLUSION

In this paper, the main attention has been made to derive a two-step optimal derivative free method of order four with three function evaluations for finding the root of a non-linear equation. Various numerical real-world application problems have been tested by proposed method and compared. It has been observed that the comparison tables does not depict in detail, for the analysis of stability and consistency of proposed method the basin of attraction of various problems have been found using the proposed method. It's observed from the above tables and basin of attraction that the proposed fourth order method is accurate, consistent and their stability is much better as compared to counterpart method with available methods in the literature. Matlab, Mathematica 2021 and maple 2021 were used to obtain the results of various problems and basin of attractions solved by various methods and compared.

ACKNOWLEDGMENT

I acknowledge the role of my supervisor and co-supervisors for this research, this research could not be possible without their guidance and support. I am also grateful to University of Sindh, Laar Campus @ Badin for allowing me to pursue my Ph.D. studies.

REFERENCES

[1] A. M. Ostrowski, *Solution of Equations in Euclidean and Banach Spaces*. London: Academic Press, 1973.

[2] A. Cordero, J. L. Hueso, E. Martínez, and J. R. Torregrosa, "Steffensen type methods for solving nonlinear equations ☆," *J. Comput. Appl. Math.*, vol. 236, no. 12, pp. 3058–3064, Jun. 2012, doi: 10.1016/j.cam.2010.08.043.

[3] Z. Xiaojian, "Modified Chebyshev-Halley methods free from second derivative," *Appl. Math. Comput.*, vol. 203, no. 2, pp. 824–827, 2008, doi: 10.1016/j.amc.2008.05.092.

[4] H. T.Kung and J. F. Truba, "Optimal Order of One-Point and Multipoint Iteration," *J. Assoc. Comput. Mach.*, vol. 21, no. 4, pp. 643–651, 1974.

[5] A. Cordero, J. L. Hueso, E. Martínez, and J. R. Torregrosa, "Generating optimal derivative free iterative methods for nonlinear equations by using polynomial interpolation," *Math. Comput. Model.*, vol. 57, no. 7–8, pp. 1950–1956, 2013, doi: 10.1016/j.mcm.2012.01.012.

[6] S. Jamali, Z. A. Kalhoro, A. W. Shaikh, and M. S. Chandio, "A New Second Order Derivative Free Method for Numerical Solution of Non-Linear Algebraic and Transcendental Equations using Interpolation Technique," *J. Mech. Contin. Math. Sci.*, vol. 16, no. 4, pp. 75–84, Apr. 2021, doi: https://doi.org/10.26782/jmcms.2021.04.00006.

[7] A. Cordero and J. R. Torregrosa, "Low-complexity root-finding iteration functions with no derivatives of any order

of convergence," *J. Comput. Appl. Math.*, vol. 275, pp. 502–515, Feb. 2015, doi: 10.1016/j.cam.2014.01.024.

[8] S. Jamali, Z. A. Kalhoro, A. W. Shaikh, and M. S. Chandio, "An Iterative, Bracketing & Derivative-Free Method for Numerical Solution of Non-Linear Equations using Stirling Interpolation Technique," *J. Mech. Contin. Math. Sci.*, vol. 16, no. 6, pp. 13–27, Jun. 2021, doi: https://doi.org/10.26782/jmcms.2021.06.00002.

[9] B. Neta, "A new derivative-free method to solve nonlinear equations," *Mathematics*, vol. 9, no. 6, pp. 1–5, 2021, doi: 10.3390/math9060583.

[10] M. S. Petković, S. Ilic, and J. Dzunic, "Derivative free two-point methods with and without memory for solving nonlinear equations," *Appl. Math. Comput.*, vol. 217, pp. 1887–1895, 2010, doi: 10.1016/j.amc.2010.06.043.

[11] F. Soleymani and S. Shateyi, "Two optimal eighth-order derivative-free classes of iterative methods," *Abstr. Appl. Anal.*, vol. 2012, 2012, doi: 10.1155/2012/318165.

[12] F. Soleymani, "Efficient optimal eighth-order derivative-free methods for nonlinear equations," *Jpn. J. Ind. Appl. Math.*, vol. 30, no. 2, pp. 287–306, Jun. 2013, doi: 10.1007/s13160-013-0103-7.

[13] S. Jamali, Z. A. Kalhoro, A. W. Shaikh, M. S. Chandio, and S. Dehraj, "A novel two point optimal derivative free method for numerical solution of nonlinear algebraic , transcendental Equations and application problems using weight function," *VFAST Trans. Math.*, vol. 10, no. 2, pp. 137–146, 2022.

[14] S. Jamali, Z. A. Kalhoro, A. W. Shaikh, M. S. Chandio, and S. Dehraj, "A new three step derivative free method using weight function for numerical solution of non-linear equations arises in application problems," *VFAST Trans. Math.*, vol. 10, no. 2, pp. 164–174, 2022.

[15] A. K. Maheshwari, "A fourth order iterative method for solving nonlinear equations," *Appl. Math. Comput.*, vol. 211, no. 2, pp. 383–391, 2009, doi: 10.1016/j.amc.2009.01.047.

[16] U. K. Qureshi, Z. A. Kalhoro, A. A. Shaikh, and S. Jamali, "Sixth Order Numerical Iterated Method of Open Methods for Solving Nonlinear Applications Problems," *Proc. Pakistan Acad. Sci. A. Phys. Comput. Sci.*, vol. 57, no. November, pp. 35–40, 2020.

[17] U. K. Qureshi, S. Jamali, Z. A. Kalhoro, and G. Jinrui, "Deprived of Second Derivative Iterated Method for Solving Nonlinear Equations," *Proc. Pakistan Acad. Sci. A. Phys. Comput. Sci.*, vol. 58, no. 2, pp. 39–44, Dec. 2021, doi: https://doi.org/10.53560/PPASA(58-2)605.

[18] U. K. Qureshi, S. Jamali, Z. A. Kalhoro, and A. G. Shaikh, "Modified Quadrature Iterated Methods of Boole Rule and Weddle Rule for Solving non-Linear Equations," *J. Mech. Contin. Math. Sci.*, vol. 16, no. 2, pp. 87–101, 2021, doi: https://doi.org/10.26782/jmcms.2021.02.00008.

[19] I. K. Argyros, M. Kansal, V. Kanwar, and S. Bajaj, "Higher-order derivative-free families of Chebyshev–Halley type methods with or without memory for solving nonlinear equations," *Appl. Math. Comput.*, vol. 315, pp. 224–245, Dec. 2017, doi: 10.1016/j.amc.2017.07.051.

[20] D. Jain, "Families of Newton-like methods with fourth-order convergence," *Int. J. Comput. Math. ISSN*, vol. 90, no. 5, pp. 1072–1082, 2013, doi: 10.1080/00207160.2012.746677.

[21] P. Sivakumar and J. Jayaraman, "Some New Higher Order Weighted Newton Methods for Solving Nonlinear Equation with Applications," *Math. Comput. Appl.*, vol. 24(2), no. 59, pp. 1–16, 2019, doi: 10.3390/mca24020059.

[22] B. Stewart, "Attractor basins of various root-finding methods," Naval Postgraduate School, 2001.

***Correspondence Author** – Sanaullah Jamali,