

Article

An Efficient Two-Step Iterative Family Adaptive with Memory for Solving Nonlinear Equations and Their Applications

Himani Sharma ¹, Munish Kansal ¹ and Ramandeep Behl ^{2,*} 

¹ School of Mathematics, Thapar Institute of Engineering and Technology, Patiala 147004, India

² Department of Mathematics, Faculty of Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia

* Correspondence: ramanbehl87@yahoo.in

Abstract: We propose a new iterative scheme without memory for solving nonlinear equations. The proposed scheme is based on a cubically convergent Hansen–Patrick-type method. The beauty of our techniques is that they work even though the derivative is very small in the vicinity of the required root or $f'(x) = 0$. On the contrary, the previous modifications either diverge or fail to work. In addition, we also extended the same idea for an iterative method with memory. Numerical examples and comparisons with some of the existing methods are included to confirm the theoretical results. Furthermore, basins of attraction are included to describe a clear picture of the convergence of the proposed method as well as that of some of the existing methods. Numerical experiments are performed on engineering problems, such as fractional conversion in a chemical reactor, Planck’s radiation law problem, Van der Waal’s problem, trajectory of an electron in between two parallel plates. The numerical results reveal that the proposed schemes are of utmost importance to be applied on various real-life problems. Basins of attraction also support this aspect.

Keywords: nonlinear equation; iterative method with memory; R -order of convergence; basin of attraction

MSC: 65H05; 65H99



Citation: Sharma, H.; Kansal, M.; Behl, R. An Efficient Two-Step Iterative Family Adaptive with Memory for Solving Nonlinear Equations and Their Applications. *Math. Comput. Appl.* **2022**, *27*, 97. <https://doi.org/10.3390/mca27060097>

Academic Editors: Sivasankaran Sivanandam and Gianluigi Rozza

Received: 5 October 2022

Accepted: 14 November 2022

Published: 18 November 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Determining the zeros of a nonlinear function promptly and accurately has become a very crucial task in many branches of science and technology. The most used technique in this regard is Newton’s method [1], which converges linearly for multiple roots and quadratically for simple roots. Various higher order schemes have also been presented in [2–8]. One amongst them is the Hansen–Patrick’s family [9] of order 3 given by

$$x_{n+1} = x_n - \left[\frac{\alpha + 1}{\alpha \pm (1 - (\alpha + 1)L_f(x_n))^{1/2}} \right] \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \dots, \quad (1)$$

where $L_f(x_n) = \frac{f''(x_n)f(x_n)}{f'^2(x_n)}$ and $\alpha \in \mathbb{R} \setminus \{-1\}$. This family comprises of Euler’s method for $(\alpha = 1)$, Ostrowski’s square-root method for $(\alpha = 0)$, Laguerre’s method for $(\alpha = \frac{1}{\nu-1}, \nu \neq 1)$ and Newton’s method as a limiting case. Despite the fact that it has cubic convergence, the involvement of second order derivative is limiting the applied region. This factor has inspired many researchers to concentrate on multipoint methods [10], since they overcome the drawbacks of one-point iterative methods with respect to the convergence order. The main motive in the development of new iterative methods is to achieve an order of convergence that is as high as possible with a certain number of functional evaluations per iteration.

Sharma et al. [11] had modified Equation (1), which is given as follows:

$$\begin{cases} y_n &= x_n - \alpha \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - \left[\frac{\beta + 1}{\beta \pm (1 - (\beta + 1)H_f(x_n))^{1/2}} \right] \frac{f(x_n)}{f'(x_n)}, \end{cases} \quad (2)$$

where $H_f(x_n) = \frac{f''(y_n)f(x_n)}{f'^2(x_n)}$ and α, β are free parameters. Here, $\beta \neq -1$. Instead of x_n , the authors calculated a second-order derivative of f at y_n . Moreover, several developments of Hansen–Patrick-type methods have been presented and examined in [12] in order to eradicate the second-order derivative. Using some appropriate approximation for $f''(x_n)$, authors in [12] presented the following method:

$$\begin{cases} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - \left[\frac{\alpha + 1}{\alpha \pm \left(\frac{f(x_n)^2 + (\beta - 2\alpha - 2)f(x_n)f(y_n) - \beta(\alpha + 1)f(y_n)^2}{f(x_n)^2 + \beta f(x_n)f(y_n)} \right)^{1/2}} \right] \frac{f(x_n)}{f'(x_n)}, \end{cases} \quad (3)$$

where α and β are free parameters. The prominent problem when using such types of methods is that they fail to work in the case where $f'(x) = 0$ and diverge or fail when the derivative is very small in the vicinity of the required root. That is why our main goal is to develop a method that is globally convergent.

On the other hand, it is sometimes possible to increase the convergence order with no further functional evaluation by making use of a self-accelerating parameter. Traub [1] referred these methods as schemes with memory as these methods use the previous information to calculate the next iterate. Traub was the first to introduce the idea of methods with memory. He made minute alterations in the already existing Steffensen method [13] and presented the first method with memory [1] as follows:

$$\begin{cases} \gamma_0, x_0 \text{ are suitably given, } w_n = x_n + \gamma_n f(x_n), 0 \neq \gamma_n \in \mathbb{R}, \\ x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, w_n]}, n = 0, 1, 2, \dots, \end{cases} \quad (4)$$

where γ_n is a self-accelerating parameter given as

$$\gamma_{n+1} = \frac{-1}{N_1'(x_n)}, \quad N_1(x) = f(x_n) + (x - x_n)f[x_n, w_n], \quad n = 0, 1, 2, \dots$$

This method has an order of convergence of 2.414. However, if we use a better self-accelerating parameter, there are apparent chances that the order of convergence will increase.

Using a secant approach, and by reusing information from the previous iteration, Traub refined a Steffensen-like method and presented the following method:

$$\begin{cases} \gamma_0 \text{ is given, } \gamma_n = \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}, n \in \mathbb{N}, \\ x_{n+1} = x_n - \frac{\gamma_n f(x_n)^2}{f(x_n + \gamma_n f(x_n)) - f(x_n)}, \end{cases} \quad (5)$$

having R-order of convergence [14] at least 2.414.

In addition, a new approach of hybrid methods is being adopted for solving nonlinear problems which can be seen in [15,16].

For finding the R-order of convergence of our proposed method with memory, we make use of the Theorem 1 given by Traub.

The rest of the paper is organized as follows. Section 2 contains the development of a new iterative method without memory and the proof of its order of convergence. Section 3 covers the inclusion of memory to develop a new iterative method with memory and its error analysis. Numerical results for the proposed methods and comparisons with some of the existing methods to illustrate our theoretical results are given in Section 4. Section 5 depicts the convergence of the methods using basins of attraction. Lastly, Section 6 presents conclusions.

Theorem 1. Suppose that (IM) is an iterative method with memory that generates a sequence $\{x_m\}$ (converging to the root ξ) of approximations to ξ . If there exists a nonzero constant ζ and nonnegative numbers s_j , $0 \leq j \leq k$, such that the inequality

$$|\epsilon_{m+1}| \leq \zeta \prod_{j=0}^k |\epsilon_{m-j}|^{s_j}$$

holds, then the R-order of convergence of the iterative method (IM) satisfies the inequality

$$O_R((IM), \xi) \geq t^*,$$

where t^* is the unique positive root of the equation

$$t^{k+1} - \sum_{j=0}^k s_j t^{k-j} = 0. \quad (6)$$

2. Iterative Method without Memory and Its Convergence Analysis

We aim to construct a new two-point Hansen–Patrick-type method without memory in this section.

Suppose $y_n = x_n - \frac{f(x_n)}{f'(x_n)}$ be the Newton's iterate. Expanding $f(y_n)$ about a point $x = x_n$ by Taylor series, we get

$$\begin{aligned} f(y_n) &\approx f(x_n) + f'(x_n)(y_n - x_n) + \frac{1}{2}f''(x_n)(y_n - x_n)^2, \\ \Rightarrow f''(x_n) &\approx \frac{2f'(x_n)^2 f(y_n)}{f(x_n)^2}. \end{aligned}$$

Further, if we expand the function $f'(y_n) = f'\left(x_n - \frac{f(x_n)}{f'(x_n)}\right)$ about $x = x_n$ by Taylor series, we have

$$\begin{aligned} f'(y_n) &\approx f'(x_n) + f''(x_n)(y_n - x_n), \\ \Rightarrow f''(x_n) &\approx \frac{f'(x_n)}{f(x_n)}(f'(x_n) - f'(y_n)). \end{aligned}$$

Using previous developments, we have

$$f''(x_n) \approx \frac{\left(\frac{2f'(x_n)^2 f(y_n)}{f(x_n)^2}\right)^2 + \left(\frac{f'(x_n)}{f(x_n)}(f'(x_n) - f'(y_n))\right)^2}{\frac{2f'(x_n)^2 f(y_n)}{f(x_n)^2} + \frac{f'(x_n)}{f(x_n)}(f'(x_n) - f'(y_n))}.$$

As we can see, this estimation for $f''(x_n)$ uses 4 functional evaluations per iteration [8], $f(x_n)$, $f(y_n)$, $f'(x_n)$ and $f'(y_n)$. To decrease the number of functional evaluations, King's approximation [17] may be used which is

$$f'(y_n) = f'(x_n) \frac{f(x_n) + \gamma f(y_n)}{f(x_n) + \beta f(y_n)},$$

when $\gamma = \beta - 2$, where β is a free parameter.

Now, using this new approximation for $f''(x_n)$ in Equation (1), authors in [12] presented the following scheme:

$$\begin{cases} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - \left[\frac{\alpha + 1}{\alpha \pm \left(\frac{f(x_n)^2 + (\beta - 2\alpha - 2)f(x_n)f(y_n) - \beta(\alpha + 1)f(y_n)^2}{f(x_n)^2 + \beta f(x_n)f(y_n)} \right)^{1/2}} \right] \frac{f(x_n)}{f'(x_n)}, \end{cases} \quad (7)$$

where α and β are free parameters.

Now, in order to extend to the method with memory, we come up with an idea of introducing a parameter b in the scheme given by Equation (7) and we present a modification in this method as follows:

$$\begin{cases} y_n &= x_n - \frac{f(x_n)}{f'(x_n) + bf(x_n)}, \\ x_{n+1} &= x_n - \left[\frac{\alpha + 1}{\alpha \pm \left(\frac{f(x_n)^2 + (\beta - 2\alpha - 2)f(x_n)f(y_n) - \beta(\alpha + 1)f(y_n)^2}{f(x_n)^2 + \beta f(x_n)f(y_n)} \right)^{1/2}} \right] \frac{f(x_n)}{f'(x_n) + bf(x_n)}, \\ n &= 0, 1, 2, \dots, \end{cases} \quad (8)$$

where α and β are free parameters.

Next, we establish the convergence results for our proposed method without memory given by Equation (8).

Convergence Analysis

Theorem 2. Suppose that $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$ be a real function suitably differentiable in a domain D . If $\xi \in D$ is a simple root of $f(x) = 0$ and an initial guess x_0 is sufficiently close to ξ , then the iterative method given by Equation (8) converges to ξ with convergence order $p = 3$ having the following error relation,

$$e_{n+1} = -\frac{1}{2}(b + a_2)(b(1 + \alpha - \beta) + (-1 + \alpha - \beta)a_2)e_n^3 + O(e_n)^4,$$

where $e_n = x_n - \xi$ and $a_n = \frac{f^{(n)}(\xi)}{n!f'(\xi)}$, $n = 2, 3, \dots$

Proof. Expanding $f(x_n)$ about $x_n = \xi$ by Taylor series, we have

$$f(x_n) = f'(\xi)(e_n + a_2e_n^2 + a_3e_n^3 + a_4e_n^4) + O(e_n)^5. \quad (9)$$

Then,

$$f'(x_n) = f'(\xi)(1 + 2a_2e_n + 3a_3e_n^2 + 4a_4e_n^3) + O(e_n)^4. \quad (10)$$

Using Equations (9) and (10), we have

$$\frac{f(x_n)}{f'(x_n) + bf(x_n)} = e_n - (b + a_2)e_n^2 + (b^2 + 2ba_2 + 2a_2^2 - 2a_3)e_n^3 + O(e_n)^4. \quad (11)$$

Using Equation (11) in the first step of Equation (8), we have

$$e_{n,y} = y_n - \xi = (b + a_2)e_n^2 + (-b^2 - 2ba_2 - 2a_2^2 + 2a_3)e_n^3 + O(e_n)^4. \quad (12)$$

Further, the Taylor's expansion of $f(y_n)$ is

$$f(y_n) = f'(\xi)(e_{n,y} + e_{n,y}^2 + e_{n,y}^3 + e_{n,y}^4) + O(e_{n,y})^5. \quad (13)$$

Using Equations (9)–(13), we have

$$\left[\frac{\alpha + 1}{\alpha \pm \left(\frac{f(x_n)^2 + (\beta - 2\alpha - 2)f(x_n)f(y_n) - \beta(\alpha + 1)f(y_n)^2}{f(x_n)^2 + \beta f(x_n)f(y_n)} \right)^{1/2}} \right] \frac{f(x_n)}{f'(x_n) + b f(x_n)} \\ = e_n + \frac{1}{2}(b + a_2)(a_2(-1 + \alpha - \beta) + b(1 + \alpha - \beta))e_n^3 + O(e_n)^4, \quad (14)$$

Finally, putting Equation (14) in the second step of Equation (8), we get

$$e_{n+1} = -\frac{1}{2}(b + a_2)(b(1 + \alpha - \beta) + (-1 + \alpha - \beta)a_2)e_n^3 + O(e_n)^4, \quad (15)$$

which is the error equation for the proposed scheme given by Equation (8) giving convergence order three. This completes the proof. \square

3. Iterative Method with Memory and Its Convergence Analysis

Now, we present an extension to the method given by Equation (8) by inclusion of memory having improved convergence order without the addition of any new functional evaluation.

If we observe clearly, it can be seen from the error relation given in Equation (15), if $b = -a_2 = -\frac{f''(\xi)}{2f'(\xi)}$, then the order of convergence of the presented scheme given by Equation (8) can possibly be improved, but this value can't be reached because the values of $f'(\xi)$ and $f''(\xi)$ are not practically available. Instead, we can use approximations calculated by already available information [18]. So, to improve the convergence order, we give an estimation using first order divided difference [19], given by $b_n = -\frac{1}{2} \frac{f'[x_n, x_{n-1}]}{f[x_n, x_{n-1}]}$, where

$f[s, t] = \frac{f(s) - f(t)}{s - t}$ denotes a first-order divided difference.

So, by replacing b by b_n in the method given by Equation (8), we obtain a new family with memory using the two previous iterations x_0, x_1 as follows:

$$\begin{cases} b_n &= -\frac{1}{2} \frac{f'[x_n, x_{n-1}]}{f[x_n, x_{n-1}]}, \\ y_n &= x_n - \frac{f(x_n)}{f'(x_n) + b_n f(x_n)}, \\ x_{n+1} &= x_n - \left[\frac{\alpha + 1}{\alpha \pm \left(\frac{f(x_n)^2 + (\beta - 2\alpha - 2)f(x_n)f(y_n) - \beta(\alpha + 1)f(y_n)^2}{f(x_n)^2 + \beta f(x_n)f(y_n)} \right)^{1/2}} \right] \\ &\quad \times \frac{f(x_n)}{f'(x_n) + b_n f(x_n)}, \quad n \in \mathbb{N}, \end{cases} \quad (16)$$

where α and β are free parameters.

Next, we establish the convergence results for our proposed method with memory given by Equation (16).

Convergence Analysis

Theorem 3. Suppose that $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$ be a real function suitably differentiable in a domain D . If $\xi \in D$ is a simple root of $f(x) = 0$ and an initial guess x_0 is sufficiently close to ξ , then the iterative method given by Equation (16) converges to ξ with convergence order at least 3.30.

Proof. Using Taylor series expansion about $x_n = \xi$, we get

$$f(x_{n-1}) = f'(\xi)(e_{n-1} + a_2 e_{n-1}^2 + a_3 e_{n-1}^3 + a_4 e_{n-1}^4 + a_5 e_{n-1}^5) + O(e_{n-1})^6, \quad (17)$$

$$f(x_n) = f'(\xi)(e_n + a_2 e_n^2 + a_3 e_n^3 + a_4 e_n^4 + a_5 e_n^5) + O(e_n)^6. \quad (18)$$

Then,

$$f'(x_{n-1}) = f'(\xi)(1 + 2a_2 e_{n-1} + 3a_3 e_{n-1}^2 + 4a_4 e_{n-1}^3 + 5a_5 e_{n-1}^4) + O(e_{n-1})^5, \quad (19)$$

$$f'(x_n) = f'(\xi)(1 + 2a_2 e_n + 3a_3 e_n^2 + 4a_4 e_n^3 + 5a_5 e_n^4) + O(e_n)^5. \quad (20)$$

Now, using previous developments, we have

$$\begin{aligned} b_n &= -\frac{1}{2} \frac{f'[x_n, x_{n-1}]}{f[x_n, x_{n-1}]} \\ &= \left(-a_2 + \left(a_2^2 - \frac{3}{2} a_3 \right) e_{n-1} + \left(-a_2^3 + \frac{5a_2 a_3}{2} - 2a_4 \right) e_{n-1}^2 + \left(a_2^4 - \frac{7}{2} a_2^2 a_3 + \frac{3a_3^2}{2} \right. \right. \\ &\quad \left. \left. + 3a_2 a_4 - \frac{5a_5}{2} \right) e_{n-1}^3 \right) + \left(\left(a_2^2 - \frac{3a_3}{2} \right) - 2 \left(a_2^3 - 2a_2 a_3 + a_4 \right) e_{n-1} + \left(3a_2^4 - \frac{17}{2} a_2^2 a_3 \right. \right. \\ &\quad \left. \left. + 3a_3^2 + 5a_2 a_4 - \frac{5a_5}{2} \right) e_{n-1}^2 \right) e_n + \left(\left(-a_2^3 + \frac{5a_2 a_3}{2} - 2a_4 \right) + \left(3a_2^4 - \frac{17}{2} a_2^2 a_3 + 3a_3^2 \right. \right. \\ &\quad \left. \left. + 5a_2 a_4 - \frac{5a_5}{2} \right) e_{n-1} \right) e_n^2 + O_3(e_{n-1} e_n). \end{aligned} \quad (21)$$

Using Equations (18), (20) and (21) in the second step of Equation (16), we get

$$\begin{aligned} y_n - \xi &= \left(\left(a_2^2 - \frac{3a_3}{2} \right) e_{n-1} + \left(-a_2^3 + \frac{5a_2 a_3}{2} - 2a_4 \right) e_{n-1}^2 \right) e_n^2 + \left(\left(\frac{a_3}{2} - 2(a_2^3 - 2a_2 a_3 \right. \right. \\ &\quad \left. \left. + a_4) \right) e_{n-1} + \frac{1}{4} (8a_2^4 - 22a_2^2 a_3 + 3a_3^2 + 20a_2 a_4 - 10a_5) e_{n-1}^2 \right) e_n^3 + O_4(e_{n-1} e_n). \end{aligned} \quad (22)$$

Then, using Equation (22) in Equation (18), we get

$$\begin{aligned} f(y_n) &= f'(\xi) \left(\left(\left(a_2^2 - \frac{3a_3}{2} \right) e_{n-1} + \left(-a_2^3 + \frac{5a_2 a_3}{2} - 2a_4 \right) e_{n-1}^2 \right) e_n^2 + \left(\frac{a_3}{2} - 2(a_2^3 \right. \right. \\ &\quad \left. \left. - 2a_2 a_3 + a_4) e_{n-1} + \frac{1}{4} (8a_2^4 - 22a_2^2 a_3 + 3a_3^2 + 20a_2 a_4 - 10a_5) e_{n-1}^2 \right) e_n^3 \right) + O_4(e_{n-1} e_n). \end{aligned} \quad (23)$$

Using Equations (18), (20)–(23) in the third step of Equation (16), we finally get

$$e_{n+1} = \left(a_2^3 - \frac{3a_2 a_3}{2} \right) e_{n-1} e_n^3 + \frac{a_2 a_3}{2} e_n^4 + O_5(e_{n-1} e_n) \quad (24)$$

Now, we can see the lowest term of the error equation is $\left(a_2^3 - \frac{3a_2 a_3}{2} \right) e_{n-1} e_n^3$, therefore, by Theorem 1, the unique positive root of the polynomial $s^2 - 3s - 1$ gives the R-order of the proposed scheme given by Equation (16), which is $s = \frac{3+\sqrt{13}}{2} \approx 3.30$. This completes our proof. \square

4. Numerical Results

Now, we will investigate the numerical results of our proposed scheme. Further, we will be comparing those results with some existing schemes, both with and without memory. All calculations have been accomplished using *Mathematica 11.1* in multiple precision arithmetic environment with the specification of a processor Intel(R) Pentium(R) CPU B960 @ 2.20 GHz (64-bit operating system) Windows 7 Ultimate @ 2009 Microsoft Corporation. We suppose that the initial value of b_0 must be selected prior to performing the iterations and a suitable x_0 be given. We have taken $b_0 = 0.01$ (or $b = 0.01$) in our computations. In all the numerical values, $Ae - h$ refers to $A \times 10^{-h}$.

We are using the following functions for our computations:

- $f_1(x) = x^2 - e^x - 3x + 2 = 0$ having one of the real zero 0.2575.

- $f_2(x) = \sin(\pi x)e^{x^2+x \cos x-1} + x \log(x \sin x + 1) = 0$ having one of the real zero 0.
- $f_3(x) = (x-2)(x^{10} + x + 2)e^{-5x} = 0$ having one of the real zero 2.
- $f_4(x) = x^2 - 1 = 0$ having one of the real zero -1 .
- $f_5(x) = \sin x = 0$ having one of the real zero 2π .

The results are computed by using the initial guesses 0.7, 0.5, 2.2, 0 and 1.69 for functions f_1, f_2, f_3, f_4 and f_5 respectively. To check the theoretical order of convergence, we have calculated the computational order of convergence [20], ρ_c (COC) using the following formula,

$$\rho_c = \frac{\log(|f(x_k)/f(x_{k-1})|)}{\log(|f(x_{k-1})/f(x_{k-2})|)}, \quad k = 2, 3, \dots,$$

considering the last three approximations in the iterative procedure. The errors of approximations to the respective zeros of the test functions, $|x_n - \xi|$ and COC are displayed in Tables 1 and 2.

Table 1. Comparison of different methods without memory.

Without Memory Methods	$ x_1 - \xi $	$ x_2 - \xi $	$ x_3 - \xi $	ρ_c	CPU Time
$f_1(x)$					
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	1.1717×10^{-4}	6.9442×10^{-15}	1.8615×10^{-35}	3.0000	0.344
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	1.0371×10^{-4}	2.8730×10^{-15}	1.8615×10^{-35}	3.0000	0.312
$PM_3(\alpha = 1, \beta = 1)$	1.1756×10^{-4}	7.0128×10^{-15}	1.8615×10^{-35}	3.0000	0.329
$HPF(\alpha = \frac{1}{2})$	7.2409×10^{-3}	2.0697×10^{-8}	4.8653×10^{-25}	2.9993	0.343
$SHM(\alpha = 1, \beta = \frac{1}{2})$	1.1696×10^{-2}	1.8718×10^{-7}	7.6275×10^{-22}	3.0008	0.344
$HM(\alpha = \frac{1}{2})$	7.2407×10^{-3}	1.8148×10^{-8}	2.8886×10^{-25}	2.9990	0.281
$f_2(x)$					
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	9.0357×10^{-3}	3.6430×10^{-7}	2.4171×10^{-20}	2.9961	0.578
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	8.7353×10^{-3}	1.5807×10^{-7}	9.6730×10^{-22}	2.9946	0.656
$PM_3(\alpha = 1, \beta = 1)$	9.0033×10^{-3}	3.5624×10^{-7}	2.2602×10^{-20}	2.9950	0.749
$HPF(\alpha = \frac{1}{2})$	1.4338×10^{-1}	5.1439×10^{-4}	6.3759×10^{-11}	2.7546	0.594
$SHM(\alpha = 1, \beta = \frac{1}{2})$	5.3590×10^{-1}	1.7758×10^{-1}	1.2161×10^{-2}	2.7518	0.843
$HM(\alpha = \frac{1}{2})$	1.6294×10^{-1}	3.4650×10^{-3}	1.2247×10^{-8}	3.1314	0.751
$f_3(x)$					
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	7.5147×10^{-5}	5.2354×10^{-17}	1.7335×10^{-53}	3.0008	0.345
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	3.0249×10^{-4}	2.6407×10^{-15}	1.8932×10^{-48}	2.9971	0.344
$PM_3(\alpha = 1, \beta = 1)$	1.8359×10^{-4}	7.8639×10^{-16}	5.8747×10^{-50}	3.0019	0.359
$HPF(\alpha = \frac{1}{2})$	6.9350×10^{-3}	3.9832×10^{-7}	7.5926×10^{-20}	2.9994	0.358
$SHM(\alpha = 1, \beta = \frac{1}{2})$	2.3532×10^{-2}	3.1713×10^{-5}	7.6632×10^{-14}	3.0020	0.328
$HM(\alpha = \frac{1}{2})$	9.4327×10^{-3}	1.0017×10^{-6}	1.2079×10^{-18}	2.9993	0.390
$f_4(x)$					
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	2.1991×10^{-1}	1.01412×10^{-3}	1.3014×10^{-10}	2.8943	0.250
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	4.0446×10^{-1}	2.5365×10^{-3}	1.0552×10^{-9}	2.7963	0.266
$PM_3(\alpha = 1, \beta = 1)$	4.0439×10^{-1}	4.4677×10^{-3}	1.1047×10^{-8}	2.7547	0.328
$HPF(\alpha = \frac{1}{2})$	F	F	F	#	-
$SHM(\alpha = 1, \beta = \frac{1}{2})$	F	F	F	#	-
$HM(\alpha = \frac{1}{2})$	F	F	F	#	-
$f_5(x)$					
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	2.4220×10^{-1}	4.9246×10^{-6}	5.9705×10^{-21}	3.1821	0.422
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	4.3566×10^{-1}	3.1927×10^{-4}	2.4757×10^{-15}	3.5597	0.406
$PM_3(\alpha = 1, \beta = 1)$	4.7757×10^{-1}	1.9651×10^{-4}	3.7695×10^{-16}	3.4779	0.328
$HPF(\alpha = \frac{1}{2})$	C	C	C	3.0017 *	-
$SHM(\alpha = 1, \beta = \frac{1}{2})$	C	C	C	3.0000 *	-
$HM(\alpha = \frac{1}{2})$	C	C	C	5.1956 *	-

F—Method fails; #—COC not required in case of failure; C—Converging to undesired root; *—COC in case of undesired root.

Table 2. Comparison of different methods with memory.

With Memory Methods	$ x_1 - \xi $	$ x_2 - \xi $	$ x_3 - \xi $	ρ_c	CPU Time
$f_1(x)$					
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	1.1718×10^{-4}	4.9506×10^{-15}	1.8615×10^{-35}	3.3435	0.407
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	1.0371×10^{-4}	2.9246×10^{-15}	1.8615×10^{-35}	3.3362	0.407
$PMM_3(\alpha = 1, \beta = 1)$	1.1756×10^{-4}	4.9995×10^{-15}	1.8615×10^{-35}	3.3434	0.344
$TM_1(\gamma_0 = 0.01)$	6.8591×10^{-3}	2.0433×10^{-7}	2.3938×10^{-18}	2.4151	0.343
$TM_2(\gamma_0 = 0.01)$	6.8591×10^{-3}	9.0200×10^{-6}	1.5202×10^{-11}	2.0037	0.312
$f_2(x)$					
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	9.0357×10^{-3}	3.1838×10^{-7}	1.6192×10^{-23}	3.6558	0.859
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	8.7353×10^{-3}	2.2053×10^{-7}	5.2810×10^{-24}	3.6119	0.875
$PMM_3(\alpha = 1, \beta = 1)$	9.0033×10^{-3}	3.1393×10^{-7}	1.5492×10^{-23}	3.6550	0.938
$TM_1(\gamma_0 = 0.01)$	2.5974×10^{-2}	2.2058×10^{-4}	1.6454×10^{-9}	2.4624	0.672
$TM_2(\gamma_0 = 0.01)$	2.5974×10^{-2}	1.1157×10^{-3}	2.4553×10^{-6}	1.9291	0.657
$f_3(x)$					
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	7.5147×10^{-5}	2.6944×10^{-14}	4.9147×10^{-47}	3.4661	0.516
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	3.0249×10^{-4}	2.5479×10^{-12}	1.6300×10^{-40}	3.4917	0.468
$PMM_3(\alpha = 1, \beta = 1)$	1.8359×10^{-4}	3.9306×10^{-13}	3.7470×10^{-43}	3.4628	0.422
$TM_1(\gamma_0 = 0.01)$	2.0344×10^{-2}	1.4543×10^{-7}	4.5676×10^{-20}	2.4298	0.313
$TM_2(\gamma_0 = 0.01)$	2.0344×10^{-2}	4.8277×10^{-5}	8.6971×10^{-11}	2.1886	0.297
$f_4(x)$					
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	2.1991×10^{-1}	2.5047×10^{-3}	3.7120×10^{-10}	3.4326	0.250
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	4.0446×10^{-1}	1.2232×10^{-2}	6.8347×10^{-8}	3.2771	0.250
$PMM_3(\alpha = 1, \beta = 1)$	4.0439×10^{-1}	1.1104×10^{-2}	5.3550×10^{-8}	3.2331	0.296
$TM_1(\gamma_0 = 0.01)$	9.9000×10^1	9.9010×10^1	4.9012×10^1	2.4357	0.359
$TM_2(\gamma_0 = 0.01)$	9.9000×10^1	6.5669×10^1	3.7895×10^1	1.9990	0.282
$f_5(x)$					
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	2.4220×10^{-1}	3.8151×10^{-3}	1.0165×10^{-10}	4.2117	0.360
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	4.3566×10^{-1}	1.1013×10^{-1}	1.1077×10^{-5}	6.8399	0.421
$PMM_3(\alpha = 1, \beta = 1)$	7.8773×10^{-1}	8.7201×10^{-2}	1.6415×10^{-5}	4.0907	0.485
$TM_1(\gamma_0 = 0.01)$	C	C	C	41.637 *	-
$TM_2(\gamma_0 = 0.01)$	C	C	C	1.7991 *	-

C—Converging to undesired root; *—COC in case of undesired root.

We have demonstrated results for our special cases without memory by taking $(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$, $(\alpha = 1, \beta = \frac{1}{2})$ and $(\alpha = 1, \beta = 1)$ in Equation (8) denoted by (PM_1) , (PM_2) and (PM_3) , respectively. Further, for our special cases with memory by taking $(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$, $(\alpha = 1, \beta = \frac{1}{2})$ and $(\alpha = 1, \beta = 1)$ in Equation (16) denoted by (PMM_1) , (PMM_2) and (PMM_3) , respectively.

For comparisons, we are considering the methods given below:

Hansen–Patrick’s family (HPF) [9]:

$$x_{n+1} = x_n - \left[\frac{\alpha + 1}{\alpha \pm (1 - (\alpha + 1)L_f(x_n))^{1/2}} \right] \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \dots, \quad (25)$$

where $L_f(x_n) = \frac{f''(x_n)f(x_n)}{f'^2(x_n)}$ and $\alpha \in \mathbb{R} \setminus \{-1\}$.

Sharma et al. method without memory (SHM) [11]:

$$\begin{cases} y_n = x_n - \alpha \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \left[\frac{\beta + 1}{\beta \pm (1 - (\beta + 1)H_f(x_n))^{1/2}} \right] \frac{f(x_n)}{f'(x_n)}, \end{cases} \quad (26)$$

where $H_f(x_n) = \frac{f''(y_n)f(x_n)}{f'^2(x_n)}$ and α and $\beta (\neq -1)$ are free parameters.

Halley's method (HM) [3]:

$$x_{n+1} = x_n - \left[1 + \frac{1}{2} \left(\frac{L_f(x_n)}{1 - \alpha L_f(x_n)} \right) \right] \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \dots, \quad (27)$$

where $L_f(x_n) = \frac{f''(x_n)f(x_n)}{f'^2(x_n)}$ and $\alpha = \frac{1}{2}$.

Traub's method with memory (TM₁) [1]:

$$\begin{cases} \gamma_0, x_0 \text{ are suitably given, } w_n = x_n + \gamma_n f(x_n), \quad 0 \neq \gamma_n \in \mathbb{R}, \\ x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \quad n = 0, 1, 2, \dots, \end{cases} \quad (28)$$

where γ_n is a self-accelerating parameter given as

$$\gamma_{n+1} = \frac{-1}{N'_1(x_n)}, \quad N_1(x) = f(x_n) + (x - x_n)f[x_n, w_n], \quad n = 0, 1, 2, \dots$$

Traub's method with memory (TM₂) [1]:

$$\begin{cases} \gamma_0 \text{ is given, } \gamma_n = \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}, \quad n \in \mathbb{N}, \\ x_{n+1} = x_n - \frac{\gamma_n f(x_n)^2}{f(x_n + \gamma_n f(x_n)) - f(x_n)}. \end{cases} \quad (29)$$

Remark 1. In Table 1, for the function $f_4(x)$, as the derivative of the function becomes zero, the existing methods HPF, SHM and HM fail. Further, for the function $f_5(x)$, HPF, SHM and HM converge to undesired root ' π '. Further, in Table 2, for the function $f_4(x)$, TM₁ and TM₂ converge to the desired root but in 11 and 14 number of iterations, respectively as we can see the errors of approximations are large in these cases. Further, for the function $f_5(x)$, TM₁ and TM₂ both converge to undesired root ' 3π '.

Further, we are considering some real life problems which are as follows:

Example 1. Firstly, we analyze the well-known Planck's radiation law problem [21],

$$\psi(\lambda) = \frac{8\pi ch_p \lambda^{-5}}{e^{\frac{ch_p}{\lambda B_k T}} - 1}, \quad (30)$$

where λ is the wavelength of radiation, h_p is the Planck's constant, T is the absolute temperature of the blackbody, c is the speed of light and B_k is the Boltzmann constant. It computes the energy density within an isothermal blackbody. We intend to obtain wavelength λ corresponding to maximum energy density $\psi(\lambda)$.

To obtain maximum value of ψ , we take $\psi'(\lambda) = 0$ which gives

$$\frac{\frac{ch_p}{\lambda B_k T} e^{\frac{ch_p}{\lambda B_k T}}}{e^{\frac{ch_p}{\lambda B_k T}} - 1} = 5. \quad (31)$$

Let $x = \frac{ch_p}{\lambda B_k T}$. Then, Equation (31) becomes

$$f_6(x) = e^{-x} + \frac{x}{5} - 1 = 0. \quad (32)$$

As we find the solutions of $f_6(x) = 0$, we get the maximum wavelength of radiation λ . As stated in [22], the L.H.S. of Equation (32) is zero when $x = 5$. Further, $e^{-5} \approx 6.738 \times 10^{-3}$. Thus, another root could appear close to $x = 5$. The desired zero is $\xi \approx 4.9651142317442763$.

Example 2. Van der Waal's equation of state [4],

$$\left(P + \frac{an^2}{V^2}\right)(V - nb) = nGT. \quad (33)$$

The following nonlinear equation needs to be solved to attain the volume V of the gas in terms of another parameters,

$$PV^3 - (nbP + nGT)V^2 + an^2V - an^2b = 0. \quad (34)$$

Here, G is the universal gas constant, P is the pressure and T is the absolute temperature. If the parameters a and b of a specific gas are given, the values of n , P and T can be calculated. Using certain values, the following nonlinear equation can be obtained,

$$f_7(x) = 0.986x^3 - 5.181x^2 + 9.067x - 5.289 = 0, \quad (35)$$

having three roots, out of which one is real and two are complex. Though our required zero is $\xi \approx 1.9298462428478622$.

Example 3. Fractional conversion in a chemical reactor [23],

$$f_8(x) = \frac{x}{1-x} - 5 \log \frac{0.4(1-x)}{0.4-0.5x} + 4.45977 = 0. \quad (36)$$

Here, x denotes the fractional conversion of quantities in a chemical reactor. If x is less than zero or greater than one, then the above fractional conversion will be of no physical meaning. Hence, x is taken to be bounded in the region of $0 \leq x \leq 1$. Moreover, the desired root is $\xi \approx 0.7573962462537538$.

Example 4. The path traversed by an electron in the air gap between two parallel plates considering the multi-factor effect is given by

$$u(t) = u_0 + \left(v_0 + c_0 \frac{E}{m\omega} \sin \omega t_0 + \beta\right)(t - t_0) + c_0 \frac{E_0}{m\omega^2} (\cos(\omega t + \beta) + \sin(\omega t + \beta)), \quad (37)$$

where u_0 and v_0 are the position and velocity of the electron at time t_0 , m and c_0 are the mass and the charge of the electron at rest and $E_0 \sin(\omega t + \beta)$ is the RF electric field between the plates. If particular parameters are chosen, Equation (37) can be simplified as

$$f_9(x) = x - \frac{1}{2} \cos x + \frac{\pi}{4} = 0. \quad (38)$$

The desired root of Equation (38) is $\xi \approx -0.3090932715417949$.

Example 5. The following nonlinear equation results from the embedment x of a sheet-pile wall,

$$f_{10}(x) = \frac{x^3 + 2.87x^2 - 10.28}{4.62} - x = 0. \quad (39)$$

The required zero of Equation (39) is $\xi \approx 2.0021$.

We have also implemented our proposed schemes given by Equations (8) and (16) on the above-mentioned problems. Tables 3 and 4 demonstrate the corresponding results. Further, Table 1 demonstrates COC for our proposed method without memory

given by Equation (8) (PM_1 , PM_2 and PM_3), method given by Equation (25) denoted by HPF , the method given by Equation (26) denoted by SHM and Halley's method given by Equation (27) denoted by HM , respectively. Table 2 demonstrates COC for our proposed method with memory given by Equation (16) (PMM_1 , PMM_2 and PMM_3), method given by Equation (28) denoted by TM_1 and the method given by Equation (29) denoted by TM_2 , respectively.

Remark 2. The proposed scheme with memory given by Equation (16) has been compared with some other methods and it is noted that our proposed scheme with memory gives better outcomes in terms of COC and errors as depicted in Tables 2 and 4. There is an obvious increase in the order of convergence.

Table 3. Comparison of different methods without memory for real-life problems.

Without Memory Methods	$ x_1 - \xi $	$ x_2 - \xi $	$ x_3 - \xi $	ρ_c	CPU Time
$f_6(x)$					
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	2.4307×10^{-6}	2.8490×10^{-16}	2.8491×10^{-16}	3.0000	0.392
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	2.2284×10^{-6}	2.8491×10^{-16}	2.8491×10^{-16}	3.0000	0.390
$PM_3(\alpha = 1, \beta = 1)$	2.5307×10^{-6}	2.8491×10^{-16}	2.8491×10^{-16}	3.0000	0.375
$HPF(\alpha = \frac{1}{2})$	1.3952×10^{-4}	1.6301×10^{-14}	2.8491×10^{-16}	3.0000	0.313
$SHM(\alpha = 1, \beta = \frac{1}{2})$	2.4927×10^{-4}	1.8505×10^{-13}	2.8491×10^{-16}	3.0000	0.328
$HM(\alpha = \frac{1}{2})$	1.4741×10^{-4}	2.0063×10^{-14}	2.8491×10^{-16}	3.0000	0.329
$f_7(x)$					
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	3.1479×10^{-3}	5.5786×10^{-7}	7.3448×10^{-18}	2.9887	0.234
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	7.3066×10^{-4}	3.5639×10^{-9}	4.1118×10^{-18}	2.9976	0.218
$PM_3(\alpha = 1, \beta = 1)$	2.1500×10^{-3}	1.7710×10^{-7}	4.2152×10^{-18}	2.9912	0.250
$HPF(\alpha = \frac{1}{2})$	4.0615×10^{-4}	1.4098×10^{-10}	4.1118×10^{-18}	3.0008	0.282
$SHM(\alpha = 1, \beta = \frac{1}{2})$	5.7189×10^{-3}	5.6997×10^{-6}	5.9525×10^{-15}	2.9774	0.234
$HM(\alpha = \frac{1}{2})$	4.5561×10^{-3}	2.3326×10^{-6}	3.3202×10^{-16}	2.9830	0.234
$f_8(x)$					
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	8.0551×10^{-4}	6.6259×10^{-8}	8.8456×10^{-17}	3.0049	0.359
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	5.9396×10^{-4}	1.3586×10^{-8}	8.8493×10^{-17}	3.0056	0.360
$PM_3(\alpha = 1, \beta = 1)$	1.0985×10^{-3}	1.7216×10^{-7}	8.7851×10^{-17}	3.0096	0.376
$HPF(\alpha = \frac{1}{2})$	5.7969×10^{-4}	4.0492×10^{-8}	8.8506×10^{-17}	3.0000	0.344
$SHM(\alpha = 1, \beta = \frac{1}{2})$	2.8780×10^{-3}	1.4099×10^{-5}	1.6623×10^{-12}	3.0246	0.313
$HM(\alpha = \frac{1}{2})$	1.9432×10^{-5}	1.0029×10^{-13}	8.8493×10^{-17}	3.0000	0.296
$f_9(x)$					
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	2.1183×10^{-3}	3.7397×10^{-10}	2.3650×10^{-30}	2.9998	0.390
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	8.4057×10^{-4}	1.0824×10^{-11}	3.0466×10^{-31}	2.9999	0.375
$PM_3(\alpha = 1, \beta = 1)$	1.8511×10^{-3}	2.4942×10^{-10}	9.1594×10^{-31}	2.9998	0.376
$HPF(\alpha = \frac{1}{2})$	1.2235×10^{-3}	1.8468×10^{-11}	3.0470×10^{-31}	2.9994	0.328
$SHM(\alpha = 1, \beta = \frac{1}{2})$	2.6486×10^{-3}	1.4723×10^{-9}	2.5407×10^{-28}	2.9996	0.406
$HM(\alpha = \frac{1}{2})$	4.1975×10^{-3}	3.6399×10^{-9}	2.3630×10^{-27}	3.0001	0.375
$f_{10}(x)$					
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	1.1429×10^{-3}	1.8779×10^{-5}	1.8779×10^{-5}	3.0002	0.298
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	6.2014×10^{-4}	1.8779×10^{-5}	1.8779×10^{-5}	3.0001	0.327
$PM_3(\alpha = 1, \beta = 1)$	1.3290×10^{-3}	1.8779×10^{-5}	1.8779×10^{-5}	3.0003	0.328
$HPF(\alpha = \frac{1}{2})$	2.6168×10^{-6}	1.8779×10^{-5}	1.8779×10^{-5}	3.0000	0.313
$SHM(\alpha = 1, \beta = \frac{1}{2})$	1.5114×10^{-3}	1.8778×10^{-5}	1.8779×10^{-5}	3.0002	0.297
$HM(\alpha = \frac{1}{2})$	1.6720×10^{-3}	1.8778×10^{-5}	1.8779×10^{-5}	3.0003	0.234

Table 4. Comparison of different methods with memory for real life problems.

With Memory Methods	$ x_1 - \xi $	$ x_2 - \xi $	$ x_3 - \xi $	ρ_c	CPU Time
$f_6(x)$					
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	2.4307×10^{-6}	2.8491×10^{-16}	2.8491×10^{-16}	3.3297	0.359
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	2.2284×10^{-6}	2.8491×10^{-16}	2.8491×10^{-16}	3.3318	0.344
$PMM_3(\alpha = 1, \beta = 1)$	2.5307×10^{-6}	2.8491×10^{-16}	2.8491×10^{-16}	3.3293	0.297
$TM_1(\gamma_0 = 0.01)$	1.5389×10^{-3}	4.7636×10^{-10}	2.8491×10^{-16}	2.4006	0.282
$TM_2(\gamma_0 = 0.01)$	1.5389×10^{-3}	8.5688×10^{-8}	5.5032×10^{-16}	2.0001	0.297
$f_7(x)$					
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	3.1479×10^{-3}	1.9881×10^{-7}	4.1148×10^{-18}	3.2847	0.125
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	7.3066×10^{-4}	2.3942×10^{-9}	4.1118×10^{-18}	3.3337	0.187
$PMM_3(\alpha = 1, \beta = 1)$	2.1500×10^{-3}	6.3764×10^{-8}	4.1119×10^{-18}	3.3027	0.171
$TM_1(\gamma_0 = 0.01)$	1.8745×10^{-2}	8.6076×10^{-4}	6.4139×10^{-7}	2.2611	0.157
$TM_2(\gamma_0 = 0.01)$	1.8745×10^{-2}	2.9838×10^{-3}	9.9340×10^{-5}	1.7706	0.203
$f_8(x)$					
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	8.0551×10^{-4}	1.3900×10^{-8}	8.8493×10^{-17}	3.2477	0.390
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	5.9396×10^{-4}	5.4849×10^{-9}	8.8493×10^{-17}	3.2580	0.374
$PMM_3(\alpha = 1, \beta = 1)$	1.0985×10^{-3}	3.4510×10^{-8}	8.8493×10^{-17}	3.2322	0.297
$TM_1(\gamma_0 = 0.01)$	1.7662×10^{-3}	2.1922×10^{-5}	1.2446×10^{-10}	2.7340	0.296
$TM_2(\gamma_0 = 0.01)$	1.7662×10^{-3}	1.1307×10^{-4}	3.9994×10^{-7}	2.0345	0.344
$f_9(x)$					
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	2.1183×10^{-3}	5.2782×10^{-11}	3.0463×10^{-31}	3.3204	0.390
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	8.4057×10^{-4}	3.2315×10^{-12}	3.0463×10^{-31}	3.3363	0.391
$PMM_3(\alpha = 1, \beta = 1)$	1.8511×10^{-3}	3.5214×10^{-11}	3.0463×10^{-31}	3.3231	0.359
$TM_1(\gamma_0 = 0.01)$	3.9978×10^{-2}	8.1932×10^{-5}	2.4800×10^{-11}	2.4205	0.328
$TM_2(\gamma_0 = 0.01)$	3.9978×10^{-2}	8.3312×10^{-4}	3.8756×10^{-7}	1.9767	0.312
$f_{10}(x)$					
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	1.1429×10^{-3}	1.8779×10^{-5}	1.8779×10^{-5}	3.2964	0.218
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	6.2014×10^{-4}	1.8779×10^{-5}	1.8779×10^{-5}	3.3102	0.187
$PMM_3(\alpha = 1, \beta = 1)$	1.3290×10^{-3}	1.8779×10^{-5}	1.8779×10^{-5}	3.2930	0.234
$TM_1(\gamma_0 = 0.01)$	2.3352×10^{-2}	4.2977×10^{-5}	1.8779×10^{-5}	2.5636	0.235
$TM_2(\gamma_0 = 0.01)$	2.3352×10^{-2}	5.4260×10^{-4}	1.9035×10^{-5}	2.0023	0.220

5. Basins of Attraction

The basins of attraction of the root t^* of $u(t) = 0$ are the sets of all initial points t_0 in the complex plane that converge to t^* on the application of the given iterative scheme. Our objective is to make use of basins of attraction to examine the comparison of several root-finding iterative methods in the complex plane in terms of convergence and stability of the method.

On this front, we have taken a 512×512 grid of the rectangle $S = [-2, 2] \times [-2, 2] \subset \mathbb{C}$. A color is assigned to each point $t_0 \in S$ on the basis of the convergence of the corresponding method starting from t_0 to the simple root and if the method diverges, a black color is assigned to that point. Thus, distinct colors are assigned to the distinct roots of the corresponding problem. It is decided that an initial point t_0 converges to a root t^* when $|t^* - t_0| < 10^{-4}$. The point t_0 is said to belong to the basins of attraction of t^* . Likewise, the method beginning from the initial point t_0 is said to diverge if no root is located in a maximum of 25 iterations. We have used MATLAB R2021a software [24] to draw the presented basins of attraction.

Furthermore, Tables 5 and 6 list the average number of iterations denoted by Avg_Iter, percentage of non-converging points denoted by P_{NC} and the total CPU time taken by the methods to generate the basins of attraction.

To carry out the desired comparisons, we have considered the test problems given below:

Table 5. Comparison of different methods without memory in terms of Avg_Iter, P_{NC} and CPU time.

Without Memory Methods	Avg_Iter	P_{NC}	CPU Time
$p_1(z)$			
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	2.7165	0	4.0501
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	2.2822	0	3.5181
$PM_3(\alpha = 1, \beta = 1)$	2.6060	0	3.8734
$HPF(\alpha = \frac{1}{2})$	2.5187	0.004×10^{-3}	2.7963
$SHM(\alpha = 1, \beta = \frac{1}{2})$	2.5187	0.004×10^{-3}	3.0479
$p_2(z)$			
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	3.1603	0	5.6380
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	2.7433	0	4.9187
$PM_3(\alpha = 1, \beta = 1)$	2.8910	0	5.0015
$HPF(\alpha = \frac{1}{2})$	2.2958	0.004×10^{-3}	2.7867
$SHM(\alpha = 1, \beta = \frac{1}{2})$	7.1570	1.505×10^{-1}	7.9437
$p_3(z)$			
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	3.9678	1.323×10^{-2}	7.2858
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	3.4784	8.185×10^{-3}	6.3408
$PM_3(\alpha = 1, \beta = 1)$	3.6042	7.816×10^{-3}	6.7897
$HPF(\alpha = \frac{1}{2})$	2.9863	0.004×10^{-3}	3.4503
$SHM(\alpha = 1, \beta = \frac{1}{2})$	8.9566	1.135×10^{-1}	10.4044
$p_4(z)$			
$PM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	3.5381	0	6.2320
$PM_2(\alpha = 1, \beta = \frac{1}{2})$	3.3973	0	6.0172
$PM_3(\alpha = 1, \beta = 1)$	3.4684	0	6.0730
$HPF(\alpha = \frac{1}{2})$	3.4316	0	3.7895
$SHM(\alpha = 1, \beta = \frac{1}{2})$	4.7686	0.304×10^{-3}	6.3023

Table 6. Comparison of different methods with memory in terms of Avg_Iter, P_{NC} and CPU time.

With Memory Methods	Avg_Iter	P_{NC}	CPU Time
$p_1(z)$			
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	2.7089	0	5.2390
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	2.4002	0	4.6904
$PMM_3(\alpha = 1, \beta = 1)$	2.5916	0	5.1093
$TM_1(\gamma_0 = 0.01)$	4.3642	1.949×10^{-3}	3.2522
$TM_2(\gamma_0 = 0.01)$	8.3338	1.390×10^{-1}	5.6000
$p_2(z)$			
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	3.1132	0	6.8290
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	2.7755	0	6.2775
$PMM_3(\alpha = 1, \beta = 1)$	2.8498	0	6.3416
$TM_1(\gamma_0 = 0.01)$	6.0252	1.110×10^{-3}	5.2045
$TM_2(\gamma_0 = 0.01)$	11.9211	2.765×10^{-1}	9.6673
$p_3(z)$			
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	3.6980	0.030×10^{-3}	8.2979
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	3.3119	0	7.5358
$PMM_3(\alpha = 1, \beta = 1)$	3.3200	0	7.7738
$TM_1(\gamma_0 = 0.01)$	8.3110	4.734×10^{-2}	6.7431
$TM_2(\gamma_0 = 0.01)$	15.0794	4.084×10^{-1}	11.8306
$p_4(z)$			
$PMM_1(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$	3.7034	0	8.2241
$PMM_2(\alpha = 1, \beta = \frac{1}{2})$	3.6684	0	8.2776
$PMM_3(\alpha = 1, \beta = 1)$	3.6692	0	8.2856
$TM_1(\gamma_0 = 0.01)$	5.4714	1.490×10^{-3}	4.5927
$TM_2(\gamma_0 = 0.01)$	9.1054	7.957×10^{-2}	7.0686

Problem 1. The first function considered is $p_1(z) = z^2 + 1$. The roots of this function are $i, -i$. The basins corresponding to our proposed method and the mentioned existing methods are shown in Figures 1 and 2. It is observed that $PM_1, PM_2, PM_3, PMM_1, PMM_2$ and PMM_3 converge to the root with no diverging points but HPF, SHM, TM_1 and TM_2 have some points painted as black.

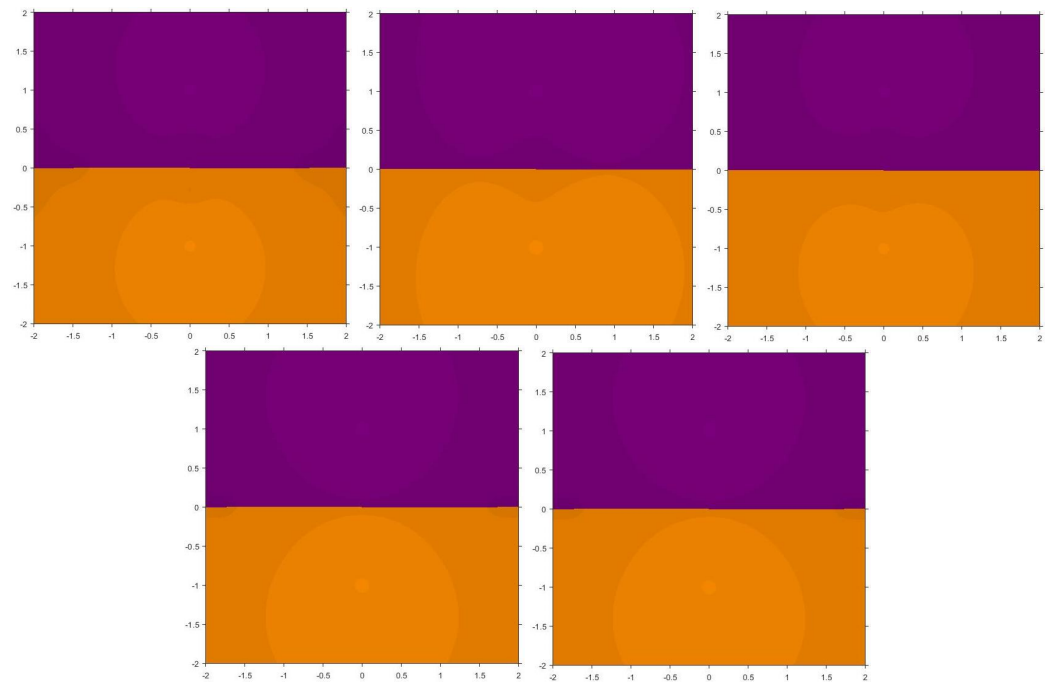


Figure 1. Basins of attraction for PM_1 , PM_2 , PM_3 , HPF , SHM , respectively for $p_1(z)$.

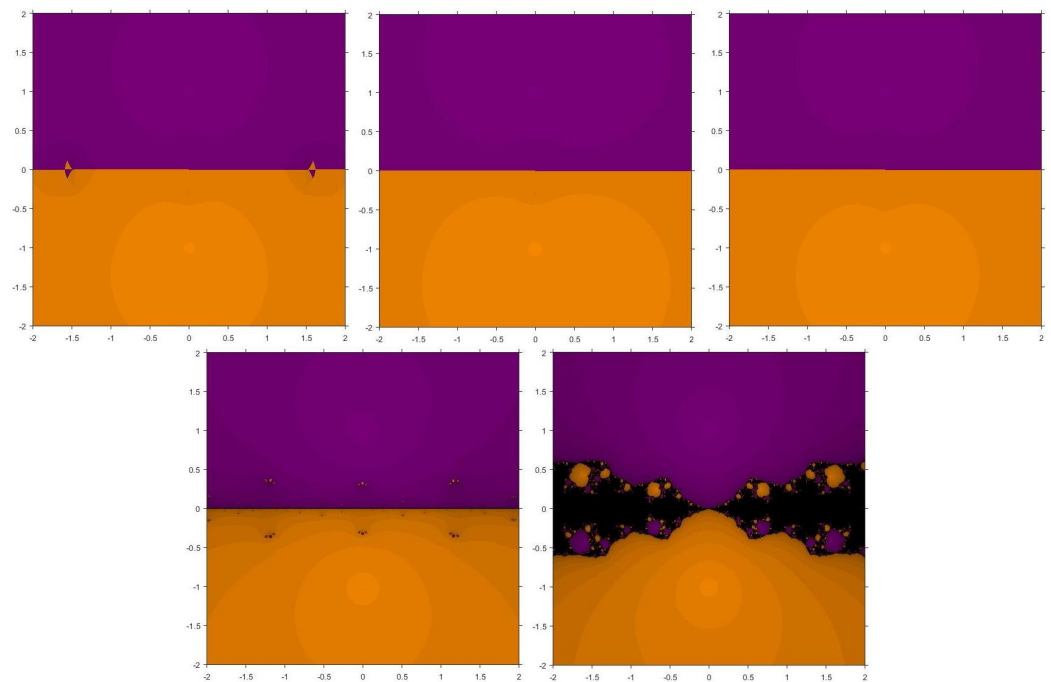


Figure 2. Basins of attraction for PMM_1 , PMM_2 , PMM_3 , TM_1 , TM_2 , respectively for $p_1(z)$.

Problem 2. Second function taken is $p_2(z) = z^3 + 1$ having roots $-1, 0.5 + 0.866i, 0.5 - 0.866i$. Figures 3 and 4 show the basins for $p_2(z)$ in which it can be seen that SHM , TM_1 and TM_2 have wider regions of divergence.

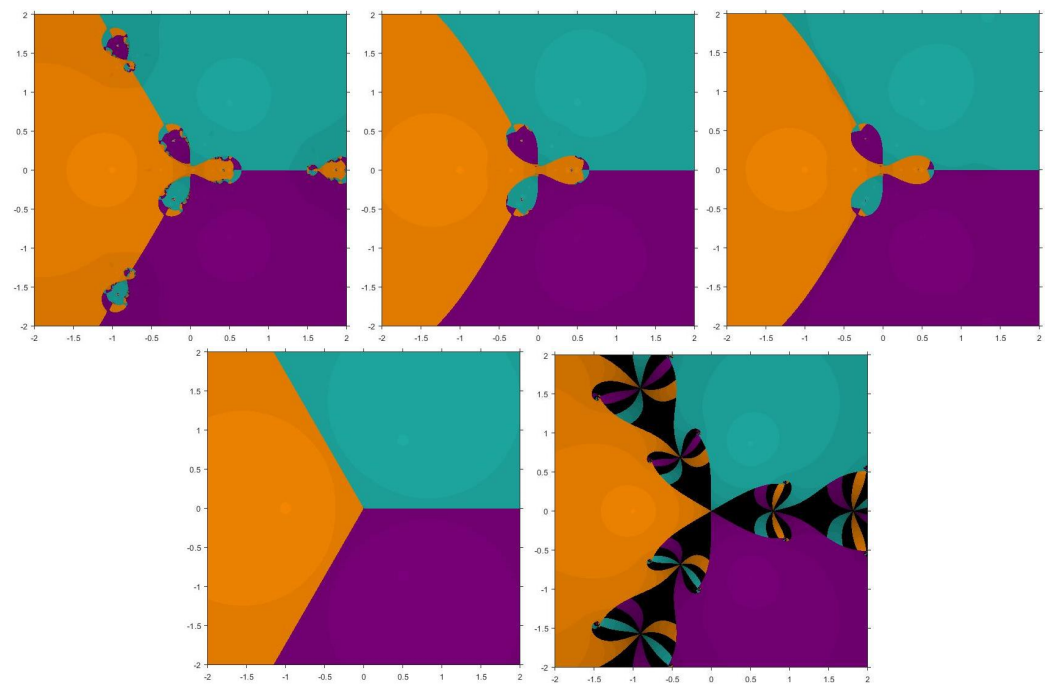


Figure 3. Basins of attraction for PM_1 , PM_2 , PM_3 , HPF , SHM , respectively for $p_2(z)$.

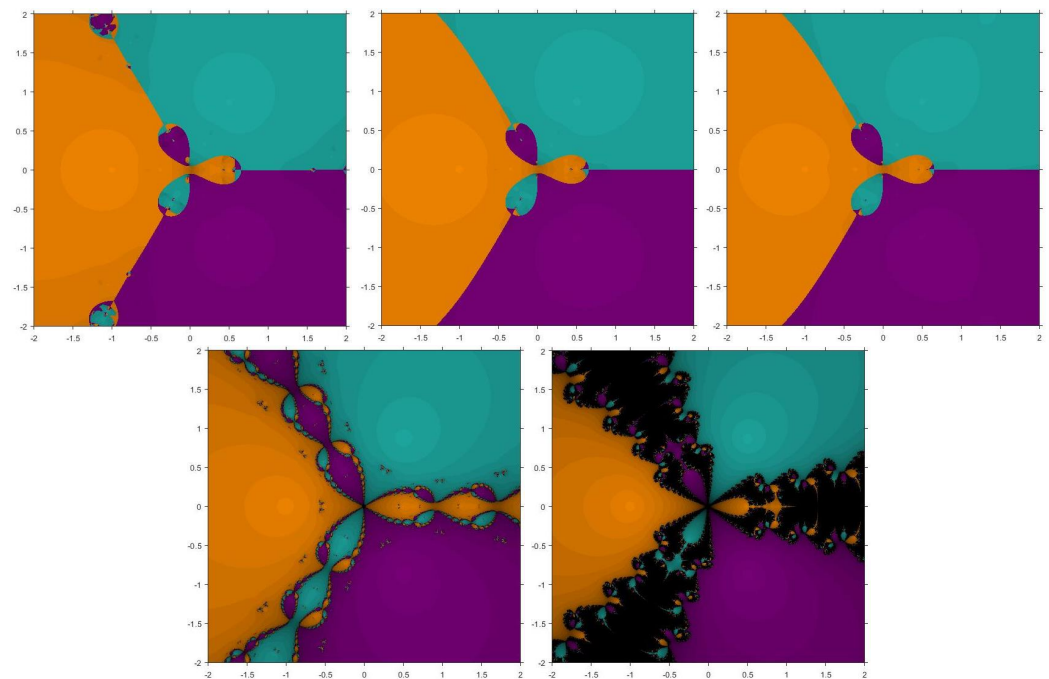


Figure 4. Basins of attraction for PMM_1 , PMM_2 , PMM_3 , TM_1 , TM_2 , respectively for $p_2(z)$.

Problem 3. The third function considered is $p_3(z) = z^4 - 1$ having roots $\pm 1, \pm i$. Figures 5 and 6 show that SHM and TM_2 have smaller basins. Although PM_1 , PM_2 , PM_3 and PMM_1 have some diverging points, yet they converge faster than the existing methods.

Problem 4. The fourth function we have taken is $p_4(z) = z^5 - z$ whose roots are $0, \pm 1, \pm i$. Figures 7 and 8 show that PM_1 , PM_2 , PM_3 , HPF , PMM_1 , PMM_2 and PMM_3 depict convergence to the root for any initial point as they have no diverging points.

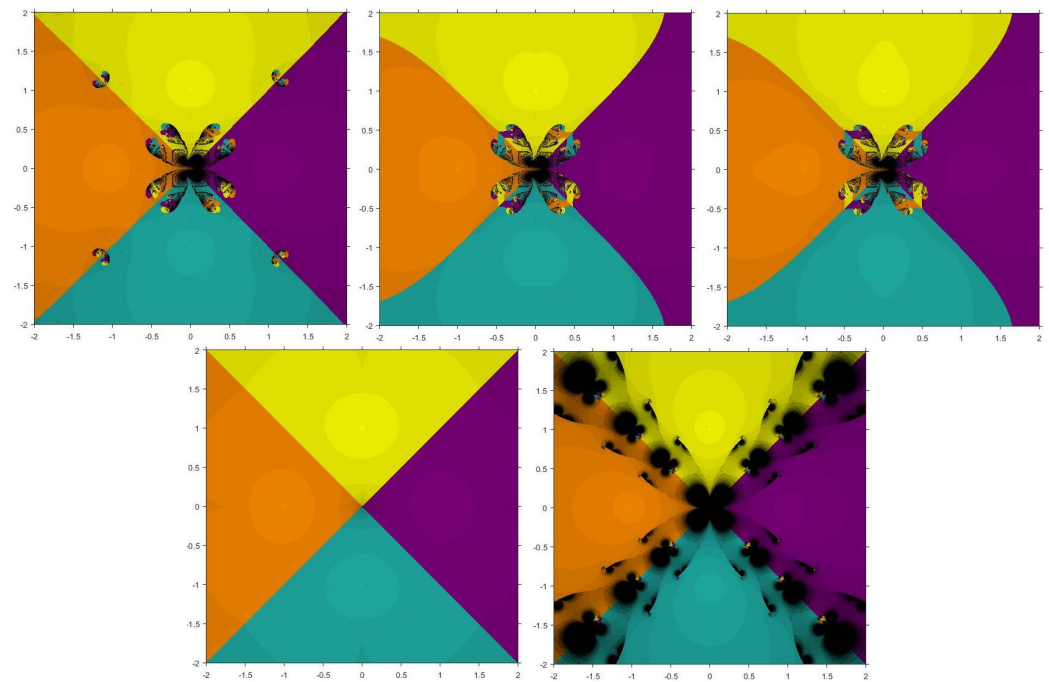


Figure 5. Basins of attraction for PM_1 , PM_2 , PM_3 , HPF , SHM , respectively for $p_3(z)$.

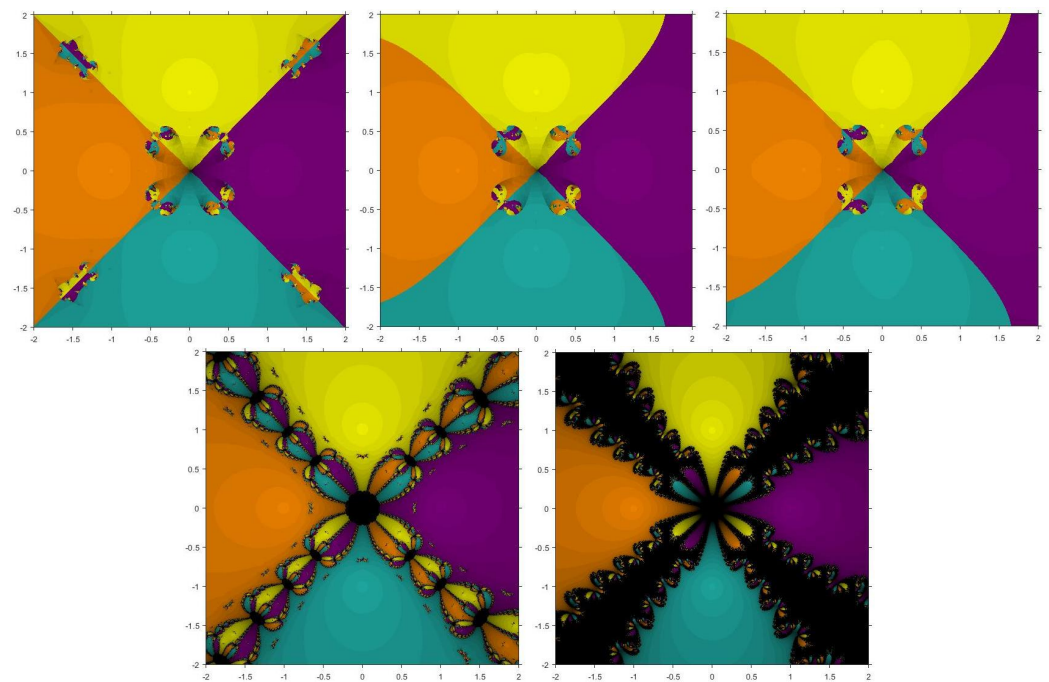


Figure 6. Basins of attraction for PMM_1 , PMM_2 , PMM_3 , TM_1 , TM_2 , respectively for $p_3(z)$.

Remark 3. One can see from Figures 1–8 and Tables 5 and 6 that there is a marginal increase in the average number of iterations per point of the existing methods, as they have more number of divergent points than that of the proposed method. Special mention to the fact that our proposed with memory method has negligible number of divergent points in the specified mesh of points and hence, larger basins of attraction. Consequently, our proposed method with memory shows faster convergence in comparison to the existing methods.

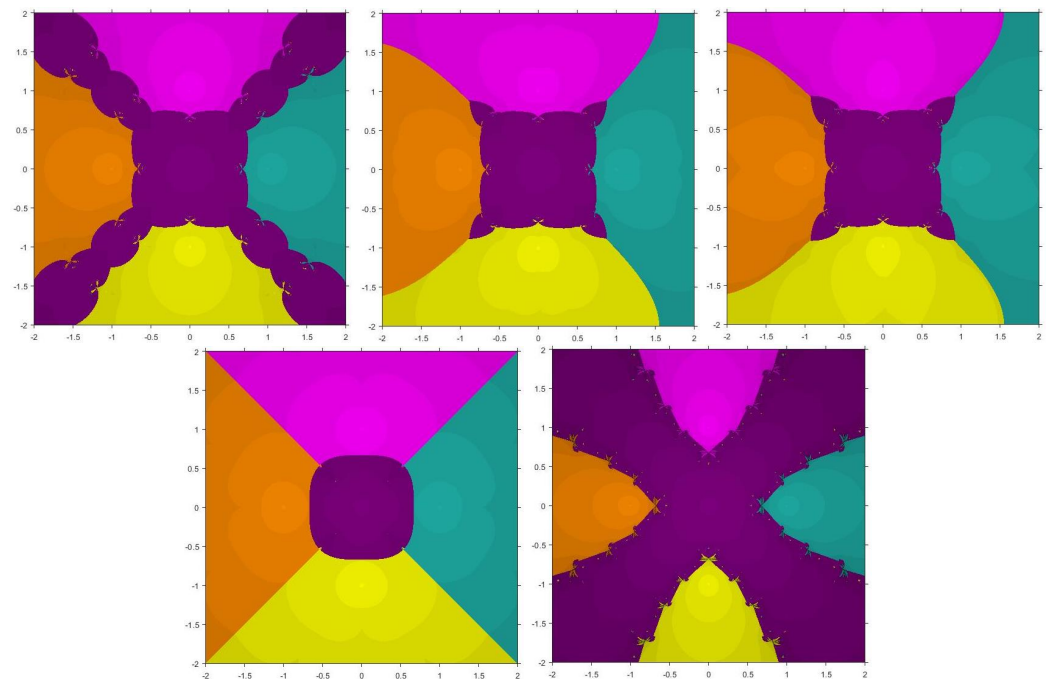


Figure 7. Basins of attraction for PM_1 , PM_2 , PM_3 , HPF , SHM , respectively for $p_4(z)$.

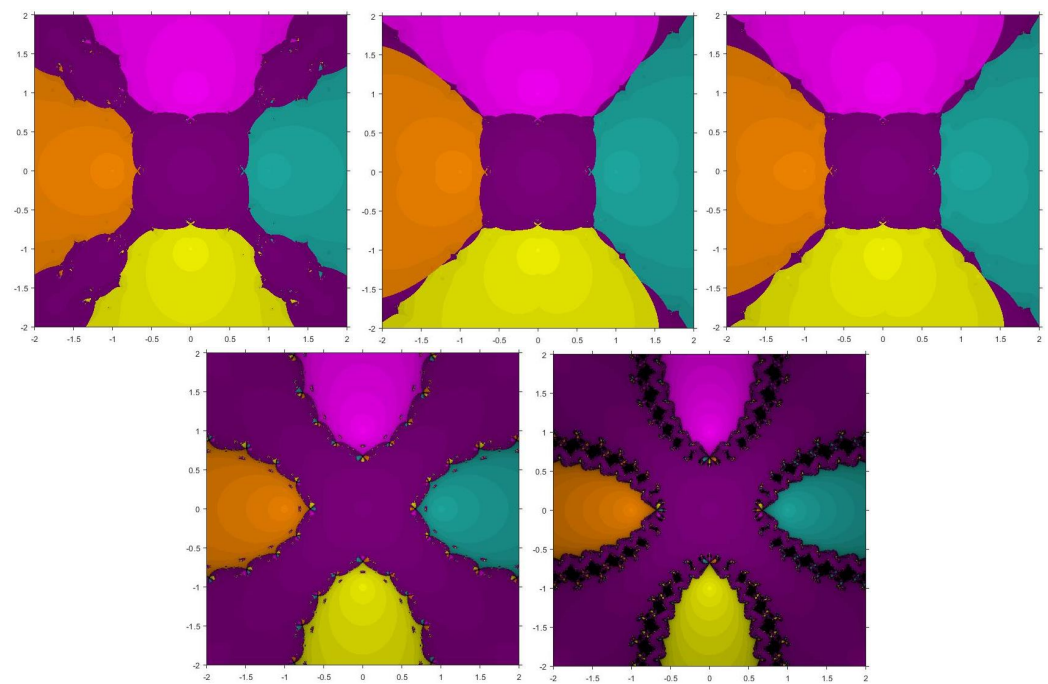


Figure 8. Basins of attraction for PMM_1 , PMM_2 , PMM_3 , TM_1 , TM_2 , respectively for $p_4(z)$.

6. Conclusions

A new method with memory has been introduced. The proposed method has a higher order of convergence in comparison with the Hansen–Patrick and Traub methods. For verification, we have carried out numerical experiments on a few test functions and some real-life problems. It is clearly visible from our results that the proposed method improves the convergence order. This increase in the convergence order has been achieved with no additional functional evaluation. Furthermore, we have also presented the basins of attraction for the proposed, as well as some existing methods, and our results point to the very fact that our proposed method converges largely to the desired zeros over a specified

region much faster. Finally, to conclude, we would say that the proposed method can be significantly used for solving nonlinear equations.

Author Contributions: M.K.: Conceptualization; Methodology; Validation; H.S.: Writing—Original draft preparation; M.K. and R.B.: Writing—Review and Editing, Supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia, under grant no. (KEP-MSc-58-130-1443). The authors, therefore, acknowledge with thanks DSR for technical and financial support.

Acknowledgments: This project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia, under grant no. (KEP-MSc-58-130-1443). The authors, gratefully acknowledge DSR for technical and financial support. Technical support provided by the Seed Money Project (TU/DORSP/57/7290) by TIET, Punjab, India is also acknowledged with thanks.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Traub, J.F. *Iterative Methods for the Solution of Equations*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1964.
2. Amat, S.; Busquier, S.; Gutiérrez, J.M. Geometric constructions of iterative functions to solve nonlinear equations. *J. Comput. Appl. Math.* **2003**, *157*, 197–205. [\[CrossRef\]](#)
3. Argyros, I.K. A note on the Halley method in Banach spaces. *Appl. Math. Comput.* **1993**, *58*, 215–224.
4. Argyros, I.K.; Kansal, M.; Kanwar, V.; Bajaj, S. Higher-order derivative-free families of Chebyshev–Halley type methods with or without memory for solving nonlinear equations. *Appl. Math. Comput.* **2017**, *315*, 224–245. [\[CrossRef\]](#)
5. Cordero, A.; Moscoso-Martinez, M.; Torregrosa, J.R. Chaos and stability in a new iterative family for solving nonlinear equations. *Algorithms* **2021**, *14*, 101. [\[CrossRef\]](#)
6. Gutiérrez, J.M.; Hernández, M.A. A family of Chebyshev–Halley type methods in Banach spaces. *Bull. Austral. Math. Soc.* **1997**, *55*, 113–130. [\[CrossRef\]](#)
7. Jain, P.; Chand, P.B. Derivative free iterative methods with memory having R-order of convergence. *Int. J. Nonlinear Sci. Numer. Simul.* **2020**, *21*, 641–648. [\[CrossRef\]](#)
8. Weerakoon, S.; Fernando, T.G.I. A variant of Newton’s method with accelerated third-order convergence. *Appl. Math. Lett.* **2000**, *13*, 87–93. [\[CrossRef\]](#)
9. Hansen, E.; Patrick, M. A family of root finding methods. *Numer. Math.* **1977**, *27*, 257–269. [\[CrossRef\]](#)
10. Petković, M.S.; Neta, B.; Petković, L.D.; Džunić, J. Multipoint methods for solving nonlinear equations: A survey. *Appl. Math. Comput.* **2014**, *226*, 635–660. [\[CrossRef\]](#)
11. Sharma, J.R.; Guha, R.K.; Sharma, R. Some variants of Hansen–Patrick method with third and fourth order convergence. *Appl. Math. Comput.* **2009**, *214*, 171–177. [\[CrossRef\]](#)
12. Kansal, M.; Kanwar, V.; Bhatia, S. New modifications of Hansen–Patrick’s family with optimal fourth and eighth orders of convergence. *Appl. Math. Comput.* **2015**, *269*, 507–519. [\[CrossRef\]](#)
13. Zheng, Q.; Li, J.; Huang, F. An optimal Steffensen-type family for solving nonlinear equations. *Appl. Math. Comput.* **2011**, *217*, 9592–9597. [\[CrossRef\]](#)
14. Ortega, J.M.; Rheinboldt, W.C. *Iterative Solutions of Nonlinear Equations in Several Variables*; Academic Press: New York, NY, USA, 1970.
15. Sihwail, R.; Solaiman, O.S.; Ariffin, K.A.Z. New robust hybrid Jarratt–Butterfly optimization algorithm for nonlinear models. *J. King Saud Univ. Comput. Inform. Sci.* **2022**, in press.
16. Sihwail, R.; Solaiman, O.S.; Omar, K.; Ariffin, K.A.Z.; Alswaitti, M.; Hashim, I. A hybrid approach for solving systems of nonlinear equations using harris hawks optimization and Newton’s method. *IEEE Access* **2021**, *9*, 95791–95807. [\[CrossRef\]](#)
17. King, R.F. A family of fourth order methods for nonlinear equations. *SIAM J. Numer. Anal.* **1973**, *10*, 876–879. [\[CrossRef\]](#)
18. Soleymani, F.; Lotfi, T.; Tavakoli, E.; Haghani, F.K. Several iterative methods with memory using self-accelerators. *Appl. Math. Comput.* **2015**, *254*, 452–458. [\[CrossRef\]](#)
19. Campos, B.; Cordero, A.; Torregrosa, J.R.; Vindel, P. Stability of King’s family of iterative methods with memory. *J. Comput. Appl. Math.* **2017**, *318*, 504–514. [\[CrossRef\]](#)
20. Jay, I.O. A note on Q-order of convergence. *BIT Numer. Math.* **2011**, *41*, 422–429. [\[CrossRef\]](#)
21. Jain, D. Families of Newton-like method with fourth-order convergence. *Int. J. Comput. Math.* **2013**, *90*, 1072–1082. [\[CrossRef\]](#)
22. Bradie, B. *A Friendly Introduction to Numerical Analysis*; Pearson Education Inc.: New Delhi, India, 2006. [\[CrossRef\]](#)
23. Shacham, M. Numerical solution of constrained nonlinear algebraic equations. *Int. J. Numer. Methods Eng.* **1986**, *23*, 1455–1481.
24. Zachary, J.L. *Introduction to Scientific Programming: Computational Problem Solving Using Maple and C*; Springer: New York, NY, USA, 2012. [\[CrossRef\]](#)