AIMS Mathematics, 9(11): 32014–32044.
DOI: 10.3934/math.20241538
Received: 15 August 2024
Revised: 30 September 2024
Accepted: 25 October 2024
Published: 11 November 2024

http://www.aimspress.com/journal/Math

Research article

Algorithms of predictor-corrector type with convergence and stability analysis for solving nonlinear systems

Dalal Khalid Almutairi[1], Ioannis K. Argyros[2], Krzysztof Gdawiec[3], Sania Qureshi[4,5], Amanullah Soomro[6], Khalid H. Jamali[6], Marwan Alquran[7] and Asifa Tassaddiq[8,*]

[1] Department of Mathematics, College of Science, Al-Zulfi Majmaah University, Al-Majmaah, 11952, Saudi Arabia

[2] Department of Computing and Mathematics Sciences, Cameron University, Lawton, OK 73505, USA

[3] Institute of Computer Science, University of Silesia, Bedzinska 39, 41-200, Sosnowiec, Poland

[4] Department of Computer Science and Mathematics, Lebanese American University, Beirut P.O. Box 13-5053, Lebanon

[5] Department of Mathematics, Near East University, 99138, Mersin, Turkey

[6] Department of Basic Sciences and Related Studies, Mehran University of Engineering & Technology, Jamshoro – 76062, Pakistan

[7] Department of Mathematics and Statistics, Jordan University of Science and Technology, P.O. Box (3030), Irbid 22110, Jordan, Jordan

[8] Department of Basic Sciences and Humanities, College of Computer and Information Sciences, Majmaah University, Al-Majmaah, 11952, Saudi Arabia

* **Correspondence:** Email: a.tassaddiq@mu.edu.sa.

Abstract: Many researchers have proposed iterative algorithms for nonlinear equations and systems of nonlinear equations; similarly, in this paper, we developed two two-step algorithms of the predictor-corrector type. A combination of Taylor's series and the composition approach was used. One of the algorithms had an eighth order of convergence and a high-efficiency index of approximately 1.5157, which was higher than that of some existing algorithms, while the other possessed fourth-order convergence. The convergence analysis was carried out in both senses, that is, local and semi-local convergence. Various complex polynomials of different degrees were considered for visual analysis via the basins of attraction. We analyzed and compared the proposed algorithms with other existing algorithms having the same features. The visual results showed that the modified algorithms had a higher convergence rate compared to existing algorithms. Real-life systems related to chemistry, astronomy, and neurology were used in the numerical simulations. The numerical simulations of the

test problems revealed that the proposed algorithms surpassed similar existing algorithms established in the literature.

Keywords: local convergence; zeros; efficiency index; Newton's algorithm; polynomiography; system of non-linear equations
Mathematics Subject Classification: 65H04, 65H05, 26C10, 30C15

1. Introduction

In the field of applied mathematics, solving algebraic and non-algebraic nonlinear equations is an open challenge for mathematicians and scientists. The exact solution of many nonlinear equations is nearly impossible. Therefore, it is important to develop new algorithms for dealing with such equations. Hence, researchers always keep searching for new numerical algorithms for getting the approximate solution of the nonlinear equations of the following type (if taken in univariate case):

$$\psi(x) = 0, \tag{1.1}$$

where ψ is a Fréchet-differentiable operator defined on a nonempty, open convex subset $\mathcal{S}$ of a Banach space $\mathcal{B}$ with values in a Banach space $\mathcal{B}'$. Similarly, if taken into the multivariate case (a system of n nonlinear equations in n unknowns) then one will have the following structure in its vector form:

$$\boldsymbol{\psi}(\mathbf{x}) = \mathbf{0}, \tag{1.2}$$

where $\boldsymbol{\psi} : \mathbb{R}^n \to \mathbb{R}^n$.

The Newton-Raphson (NR) algorithm is a common iterative algorithm for finding approximate solutions to nonlinear equations with variables that have real values [1]. The fundamental notion of this concept is centered on the use of linear approximations. When presented with an initial estimate for the root of a function, denoted as $\psi(x) = 0$, it is possible to create a linear approximation of the function in the vicinity of this estimate by employing the Taylor series. This approximation facilitates the formulation of the subsequent estimate for the root, which is then iteratively revised to approximate the solution. The algorithm commences by establishing an initial approximation and thereafter iteratively enhances it through the evaluation of the function and its derivative. The aforementioned approach is notable for its expeditious convergence in close proximity to the true root, its simplicity, and its wide-ranging applicability across functions.

Nevertheless, the NR approach does have its limitations. It is worth noting that the convergence of the algorithm is not always assured and is strongly dependent on the initial estimate. The computational demand or complexity of computing the derivative at each iteration can be significant for certain functions. Moreover, it is worth noting that the procedure may encounter difficulties or deviate from its intended path when launched in proximity to an inflection point or when the derivative approaches zero. When confronted with a function that has numerous roots, the outcome of the approach can differ, as it may converge to any of these roots depending on the initial approximation. To enhance its effectiveness, it is crucial to carefully choose the initial estimate and be aware of the potential constraints of the procedure. Adjustments, like introducing a damping factor, can also be incorporated

to enhance its robustness. The classical Newton's technique to find the approximate solution of the non-linear equation (1.1) is given as:

$$x_{j+1} = x_j - \frac{\psi(x_j)}{\psi'(x_j)}, \quad j = 1, 2, \ldots. \tag{1.3}$$

This second-order one-step approach has two function evaluations per iteration. The theoretical limit of the one-point techniques is beaten by the multi-step iterative approaches of both higher convergence order and better efficiency index. The formula for the efficiency index is $\kappa = r^{1/p}$, where r is the order of the algorithm and p is the number of function evaluations per iteration. The authors proposed solving a system of two equations to solve a scalar problem in [2]. So, the linked system's solutions on the identity line solve the problem. In problematic circumstances where the scalar Newton approach fails, this strategy succeeds. The authors in [3], proposed an alternate algorithm for solving nonlinear equation systems when Newton's algorithm fails. Based on the paper [2], the proposed technique extends systems. The theory behind Newton's algorithm is used to solve an associated system to approximate a system of equations. Many such iterative algorithms have been published.

Researchers have focused on improving the convergence order and reducing function evaluations, leading to the development of multi-step iterative algorithms. The authors introduced a three-step iterative nonlinear approach for nonlinear equations and systems in the paper [4]. Iterations of the proposed technique need three function evaluations and two first-order derivative checks. The theory proves the proposed strategy is sixth-order convergent. The suggested algorithm is compared to others based on error distributions, computational efficiency, and CPU times. Qureshi et al. in [5] combined existing algorithms to create an optimal fourth-order algorithm for solving scalar and vector forms of problems. A new root-finding algorithm [6] that combines forward and finite-difference methods, is efficient, derivative-free, and cheaper per iteration. Quantitative and graphical studies reveal that the approach is quintic-order convergent and outperforms previous algorithms. An eighth-order three-step algorithm after merging second-order NR with an existing fourth-order algorithm has been devised in [7] that is later proven to be efficient while solving single-variable nonlinear physical models. In a research study [8], a twelfth-order numerical algorithm was proposed to solve the nonlinear equations of the type (1.1), however, the algorithm was four-step and thus required seven function evaluations per iteration. Algorithms for solving nonlinear algebraic equations are also needed while discretizing ordinary and partial differential equations [9–12].

The following are some of the well-known iterative algorithms having fourth- and eighth-order convergence selected for comparison with algorithms developed in this research work. For example, Hueso et al. in [13] proposed an iterative fourth-order algorithm, i.e., Jarratt's method. Newton's step is used as a predictor of the following form and abbreviated as PJNM:

$$\begin{aligned} y_j &= x_j - \frac{2}{3}\frac{\psi(x_j)}{\psi'(x_j)}, \\ x_{j+1} &= x_j - \frac{\psi(x_j)}{\psi'(x_j)} \cdot \frac{3\psi'(y_j) + \psi'(x_j)}{6\psi'(y_j) - 2\psi'(x_j)}, \end{aligned} \tag{1.4}$$

where $j = 1, 2, \ldots$.

Behl in 2015 [14] constructed an optimal algorithm of fourth-order convergence of the following

form and abbreviated as KTNM:

$$\begin{aligned} y_j &= x_j - \frac{\psi(x_j)}{\psi'(x_j)}, \\ x_{j+1} &= y_j - \frac{\psi(y_j)}{\psi'(x_j)} \cdot \frac{\psi(x_j) + 2\psi(y_j)}{\psi(x_j)}, \end{aligned} \tag{1.5}$$

where $j = 1, 2, \ldots$.

Similarly, Özban and Kaya in [15] proposed two iterative fourth-order algorithms of the following form and abbreviated as CNM1 and CNM2:

$$\begin{aligned} y_j &= x_j - \frac{2\psi(x_j)}{3\psi'(x_j)}, \\ x_{j+1} &= x_j - \frac{16\psi'(y_j)^2}{-9\psi'(x_j)^2 + 22\psi'(x_j)\psi'(y_j) + 3\psi'(y_j)^2} \cdot \frac{\psi(x_j)}{\psi'(y_j)}, \end{aligned} \tag{1.6}$$

and

$$\begin{aligned} y_j &= x_j - \frac{2\psi(x_j)}{3\psi'(x_j)}, \\ x_{j+1} &= x_j - \frac{5\psi'(x_j)^2 - 12\psi'(x_j)\psi'(y_j) + 15\psi'(y_j)^2}{8\psi'(y_j)^2} \cdot \frac{\psi(x_j)}{\psi'(y_j)}, \end{aligned} \tag{1.7}$$

where $j = 1, 2, \ldots$.

In 2021, Kong-ied proposed a new eight-order algorithm [16], abbreviated as ONM:

$$\begin{aligned} y_j &= x_j - \frac{\psi(x_j)}{\psi'(x_j)}, \\ z_j &= y_j - \frac{\psi(x_j)^2\psi(y_j)}{\psi(x_j)^2\psi'(x_j) - 2\psi(x_j)\psi'(x_j)\psi(y_j) + \psi'(x_j)\psi(y_j)^2}, \\ x_{j+1} &= z_j - \frac{\psi(z_j)}{\psi'(z_j)}, \end{aligned} \tag{1.8}$$

where $j = 1, 2, \ldots$.

Motivated by the current studies in this direction, we attempt to propose two algorithms having eighth- and fourth-order convergence for solving nonlinear models of the type (1.1). The development of the algorithms was extensively supported by alternating Taylor's series expansion and the classical Halley algorithm. Moreover, the developed algorithms are equally applicable to both univariate and multivariate cases.

This article is structured as follows: In Section 2, we present the basic construction of both algorithms including their local convergence analysis based on Taylor's expansion, while Section 3 presents the local and semilocal convergence analysis for the proposed algorithms under consideration. A detailed visual analysis via the polynomiography of the algorithms is presented in Section 4. To prove the better performance of the proposed algorithms, some numerical experiments (single-variable and system), including both academic and real-life situations, are conducted in Section 5, whereas the concluding remarks with some future directions are described in Section 6.

2. Derivation of the proposed algorithms

In this section, we present the construction of the proposed two algorithms. Moreover, we perform the convergence analysis of the algorithms via Taylor's expansion.

2.1. Construction of the eighth-order algorithm

Suppose that ψ is a real function defined on the interval $W \subset \mathbb{R}$. Let $\xi \in W$ be an exact simple zero for the non-linear equation $\psi(x) = 0$ and further let x_j be an initial guess close to the exact root of the nonlinear equation. By the Taylor series expansion, the function ψ is expanded to three terms around the point x_j as follows:

$$\psi(x) = \psi(x_j) + (x - x_j)\psi'(x_j) + \frac{(x - x_j)^2}{2!}\psi''(x_j) + O(x - x_j)^3. \tag{2.1}$$

For getting the next approximation x_{j+1} for the root of $\psi(x) = 0$ in Eq (2.1), we assume that $\psi(x_{j+1}) = 0$. Hence, we get

$$\psi(x_j) + (x_{j+1} - x_j)\psi'(x_j) + \frac{(x_{j+1} - x_j)^2}{2!}\psi''(x_j) = 0. \tag{2.2}$$

After simplification, we get

$$x_{j+1} = x_j - \frac{\psi(x_j)}{\psi'(x_j)} - \frac{(x_{j+1} - x_j)^2}{2}\frac{\psi''(x_j)}{\psi'(x_j)}. \tag{2.3}$$

From [17], we have

$$x_{j+1} - x_j = -\frac{2\psi(x_j)\psi'(x_j)}{2\psi'^2(x_j) - \psi(x_j)\psi''(x_j)}. \tag{2.4}$$

Substituting Eq (2.4) into Eq (2.3), we get

$$x_{j+1} = x_j - \frac{\psi(x_j)}{\psi'(x_j)} - \frac{\left(-\dfrac{2\psi(x_j)\psi'(x_j)}{2\psi'^2(x_j) - \psi(x_j)\psi''(x_j)}\right)^2}{2} \cdot \frac{\psi''(x_j)}{\psi'(x_j)}. \tag{2.5}$$

After simplification, we get

$$x_{j+1} = x_j - \left[\frac{\psi(x_j)}{\psi'(x_j)} + \frac{2\psi^2(x_j)\psi'(x_j)\psi''(x_j)}{[2\psi'^2(x_j) - \psi(x_j)\psi''(x_j)]^2}\right]. \tag{2.6}$$

Using the second-order Newton-Raphson algorithm as a predictor and the Eq (2.6) as a corrector, we develop a predictor-corrector (PC) type algorithm as given below:

$$\begin{aligned} y_j &= x_j - \frac{\psi(x_j)}{\psi'(x_j)}, \\ x_{j+1} &= y_j - \left[\frac{\psi(y_j)}{\psi'(y_j)} + \frac{2\psi^2(y_j)\psi'(y_j)\psi''(y_j)}{[2\psi'^2(y_j) - \psi(y_j)\psi''(y_j)]^2}\right], \end{aligned} \tag{2.7}$$

where $j = 1, 2, \ldots$. The new PC numerical algorithm given by (2.7) has an eighth-order of convergence and is denoted as PCNM8. The proposed PCNM8 algorithm has five function evaluations

(two evaluations of functions, two first-order derivatives, and one second-order derivative), and the efficiency index of this algorithm is $r^{1/5} \approx 1.5157$, where r stands for the order of the algorithm. It may be noted that the efficiency index of the proposed eighth-order algorithm is higher than the classical NR algorithm.

2.2. Construction of the fourth-order algorithm

In several research studies, researchers have employed different strategies, including finite differences, weight functions, regularization, and the secant algorithm, among others; to get rid of higher-order derivatives. Following the same approach, we attempt to use several possible approaches to remove the second-order derivative in the above-developed eighth-order algorithm in Eq (2.7). For this purpose, the following structure:

$$\psi''(y_j) = \frac{3}{2} \cdot \frac{\psi'(x_j) - \psi'(y_j)}{\psi'(x_j)}. \tag{2.8}$$

for the second derivative taken from [18] is chosen due to its simplicity and better simulation results.

Substituting Eq (2.8) into Eq (2.7), we get

$$\begin{aligned} y_j &= x_j - \frac{\psi(x_j)}{\psi'(x_j)}, \\ x_{j+1} &= y_j - \left[\frac{\psi(y_j)}{\psi'(y_j)} - \frac{12\psi^2(y_j)\psi'(y_j)\psi'(x_j)(\psi'(y_j) - \psi'(x_j))}{\left(4\psi'^2(y_j)\psi'(x_j) + 3\psi(y_j)\psi'(y_j) - 3\psi(y_j)\psi'(x_j)\right)^2} \right]. \end{aligned} \tag{2.9}$$

Hence, the algorithm (2.9) is a modified algorithm that is now free from the second derivative and abbreviated as PCNM4.

2.3. Convergence analysis of proposed algorithms via Taylor expansion

Theorem 1. *Let $\xi \in W$ be a simple root of a differential function $\psi : W \subset \mathbb{R} \to \mathbb{R}$, where W is an open interval. If x_0 is an initial guess considerably near ξ, then the modified algorithm defined by* (2.7) *has eighth-order convergence and satisfies the following error equation:*

$$e_{j+1} = \frac{3\psi''(\xi)^7}{64\psi'(\xi)^7} e_j^8 + O(e_j^9), \tag{2.10}$$

where $e_j = x_j - \xi$.

Proof. Since ξ is a root of ψ and $e_j = x_j - \xi$ is the error at j^{th} iteration, we can expand $\psi(x_j)$ in power of e_j by Taylor's series expansion as follows:

$$\psi(x_j) = \psi'(\xi)e_j + \frac{1}{2}\psi''(\xi)e_j^2 + O(e_j)^3. \tag{2.11}$$

By Taylor's series for $\frac{1}{\psi'(x_j)}$ about ξ, we obtain

$$\frac{1}{\psi'(x_j)} = \frac{1}{\psi'(\xi)} - \frac{\psi''(\xi)e_j}{\psi'(\xi)^2} + O(e_j)^2. \tag{2.12}$$

Multiplying (2.11) and (2.12), we get

$$\frac{\psi(x_j)}{\psi'(x_j)} = e_j - \frac{1}{\psi'(\xi)} - \frac{\psi''(\xi)e_j^2}{2\psi'(\xi)} - \frac{\psi''(\xi)^2e_j^3}{2\psi'(\xi)^2} + O(e_j)^4. \tag{2.13}$$

Using (2.13) in the first step of (2.7), we get

$$\sigma_j = \frac{\psi''(\xi)e_j^2}{2\psi'(\xi)} + \frac{\psi''(\xi)^2e_j^3}{2\psi'(\xi)^2} + O(e_j)^4, \tag{2.14}$$

where $\sigma_j = y_j - \xi$. By Taylor's series for $\psi(y_j)$ about ξ, we obtain

$$\psi(y_j) = \psi'(\xi)\sigma_j + \frac{1}{2}\psi''(\xi)\sigma_j^2 + O(\sigma_j)^3. \tag{2.15}$$

By Taylor's series for $\frac{1}{\psi'(y_j)}$ about ξ, we obtain

$$\frac{1}{\psi'(y_j)} = \frac{1}{\psi'(\xi)} - \frac{\psi''(\xi)\sigma_j}{\psi'(\xi)^2} + O(\sigma_j)^2. \tag{2.16}$$

Multiplying (2.15) and (2.16), we get

$$\frac{\psi(y_j)}{\psi'(y_j)} = \sigma_j - \frac{1}{\psi'(\xi)} - \frac{\psi''(\xi)\sigma_j^2}{2\psi'(\xi)} - \frac{\psi''(\xi)^2\sigma_j^3}{2\psi'(\xi)^2} + O(\sigma_j)^4. \tag{2.17}$$

By Taylor's series for $\psi'(y_j)$ about ξ, we obtain

$$\psi'(y_j) = \psi'(\xi) + \psi''(\xi)\sigma_j + O(\sigma_j)^2. \tag{2.18}$$

By Taylor's series for $\psi''(y_j)$ about ξ, we obtain

$$\psi''(y_j) = \psi''(\xi) + \psi''(\xi)\sigma_j + O(\sigma_j)^2. \tag{2.19}$$

Squaring (2.15), we have

$$\psi^2(y_j) = \psi'(\xi)^2\sigma_j^2 + \psi'(\xi)\psi''(\xi)\sigma_j^3 + \frac{\psi''(\xi)^2\sigma_j^4}{4} + O(\sigma_j)^5. \tag{2.20}$$

Squaring (2.18), we have

$$\psi'^2(y_j) = \psi'(\xi)^2 + 2\psi'(\xi)\psi''(\xi)\sigma_j + \psi''(\xi)^2\sigma_j^2 + O(\sigma_j)^3. \tag{2.21}$$

Using the values of $\psi^2(y_j)$, $\psi'(y_j)$ and $\psi''(y_j)$ in numerator, i.e., $2\psi^2(y_j)\psi'(y_j)\psi''(y_j)$ of the second step of (2.7), we get

$$2\sigma_j^2\psi'(\xi)^3\psi''(\xi) + 4\sigma_j^3\psi'(\xi)^2\psi''(\xi)^2 + \frac{5}{2}\sigma_j^4\psi'(\xi)\psi''(\xi)^3 + O(\sigma_j)^5. \tag{2.22}$$

Using the values of $\psi(y_j)$, $\psi'^2(y_j)$ and $\psi''(y_j)$ in denominator, i.e., $[2\psi'^2(y_j) - \psi(y_j)\psi''(y_j)]^2$ of the second step of (2.7), we get

$$2\psi'(\xi)^2 + 2\sigma_j\psi'(\xi)\psi''(\xi) + \frac{3}{2}\sigma_j^2\psi''(\xi)^2 + \sigma_j^2\psi'(\xi)\psi'''(\xi) + O(\sigma_j)^3. \tag{2.23}$$

Finally, by using (2.18), (2.22) and (2.23) in the second step of Eq (2.7), we get

$$x_{j+1} = \xi - \frac{3\sigma_j^4\psi''(\xi)^3}{4\psi'(\xi)^3} - \frac{\sigma_j^5\psi''(\xi)^4}{8\psi'(\xi)^4} + \frac{123\sigma_j^6\psi''(\xi)^5}{32\psi'(\xi)^5} + O(\sigma_j)^7. \tag{2.24}$$

By substituting $\sigma_j = \frac{\psi''(\xi)e_j^2}{2\psi'(\xi)} + \frac{\psi''(\xi)^2e_j^3}{2\psi'(\xi)^2} + O(e_j)^4$ in (2.24), we get

$$e_{j+1} = \frac{3\psi''(\xi)^7}{64\psi'(\xi)^7}e_j^8 + O(e_j)^9. \tag{2.25}$$

Hence, Eq (2.25) shows that the proposed algorithm (2.7) has eighth-order of convergence. □

As the convergence analysis carried out for the proposed eighth-order algorithm (2.7), the convergence analysis for the proposed fourth-order algorithm is addressed in the same way.

Theorem 2. *Let $\xi \in W$ be a simple root of a differential function $\psi : W \subset \mathbb{R} \to \mathbb{R}$, where W is an open interval. Let x_0 be an initial guess considerably near to the exact root ξ. Then, the modified algorithm defined by* (2.9) *has fourth-order convergence and satisfies the following error equation:*

$$e_{j+1} = \frac{\psi''(\xi)^3}{8\psi'(\xi)^3}e_j^4 + O(e_j)^5. \tag{2.26}$$

Proof. Since ξ is a root of ψ and $e_j = x_j - \xi$ is the error at j^{th} iteration, we can expand $\psi(x_j)$ in power of e_j by Taylor's series expansion as follows:

$$\psi(x_j) = \psi'(\xi)e_j + \frac{1}{2}\psi''(\xi)e_j^2 + O(e_j)^3. \tag{2.27}$$

By Taylor's series for $\frac{1}{\psi'(x_j)}$ about ξ, we obtain

$$\frac{1}{\psi'(x_j)} = \frac{1}{\psi'(\xi)} - \frac{\psi''(\xi)e_j}{\psi'(\xi)^2} + O(e_j)^2. \tag{2.28}$$

Multiplying (2.27) and (2.28), we get

$$\frac{\psi(x_j)}{\psi'(x_j)} = e_j - \frac{1}{\psi'(\xi)} - \frac{\psi''(\xi)e_j^2}{2\psi'(\xi)} - \frac{\psi''(\xi)^2e_j^3}{2\psi'(\xi)^2} + O(e_j)^4. \tag{2.29}$$

Using (2.29) in the first step of (2.9), we get

$$\sigma_j = \frac{\psi''(\xi)e_j^2}{2\psi'(\xi)} + \frac{\psi''(\xi)^2e_j^3}{2\psi'(\xi)^2} + O(e_j)^4. \tag{2.30}$$

By Taylor's series for $\psi(y_j)$ about ξ, we obtain

$$\psi(y_j) = \psi'(\xi)\sigma_j + \frac{1}{2}\psi''(\xi)\sigma_j^2 + O(\sigma_j)^3. \tag{2.31}$$

By finding the derivative of $\psi(x_j)$ and $\psi(y_j)$ respectively, we have

$$\psi'(x_j) = \psi'(\xi) + \psi''(\xi)e_j + O(e_j)^2, \tag{2.32}$$

$$\psi'(y_j) = \psi'(\xi) + \psi''(\xi)\sigma_j + O(\sigma_j)^2. \tag{2.33}$$

By Taylor's series for $\frac{1}{\psi'(y_j)}$ about ξ, we obtain

$$\frac{1}{\psi'(y_j)} = \frac{1}{\psi'(\xi)} - \frac{\psi''(\xi)\sigma_j}{\psi'(\xi)^2} + O(\sigma_j)^2. \tag{2.34}$$

Multiplying (2.31) and (2.34), we get

$$\frac{\psi(y_j)}{\psi'(y_j)} = \sigma_j - \frac{1}{\psi'(\xi)} - \frac{\psi''(\xi)\sigma_j^2}{2\psi'(\xi)} - \frac{\psi''(\xi)^2\sigma_j^3}{2\psi'(\xi)^2} + O(\sigma_j)^4. \tag{2.35}$$

By putting all the above equations into second step of (2.9), we get

$$e_{j+1} = \frac{\psi''(\xi)^3}{8\psi'(\xi)^3}e_j^4 + O(e_j)^5. \tag{2.36}$$

The error equation (2.36) shows that the proposed algorithm (2.9) has fourth-order convergence. □

The flowchart of the proposed two-step iterative algorithms is shown in Figure 1.

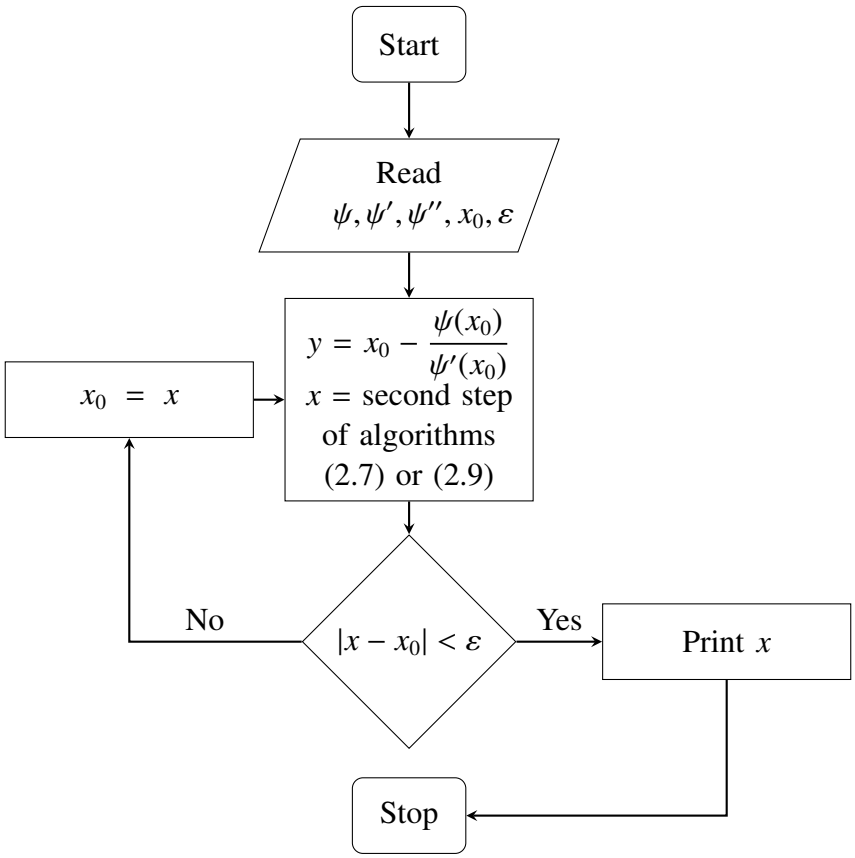


Figure 1. Flowchart of the proposed predictor–corrector algorithms.

3. Local and semi-local convergence

There are certain limitations with the local convergence analysis performed in Section 2 for the algorithm (2.7).

(L_1) The algorithm (2.7) can be also used to solve equations on the real line.
(L_2) The existence of at least fourth derivative is required to show convergence, and the solution has to be simple. Consider the example, for $W = [-1.5, 1.5]$, and function ψ defined on W by $\psi(x) = x^4 \log(x) + x^5 - x^4$ for $x \neq 0$, and $\psi(0) = 0$. Then, clearly $\zeta = 1$ solves the equation $\psi(x) = 0$, but $\psi^{(4)}$ does not exist at $x = 0$. Consequently, the results of Section 2 cannot guarantee convergence to ζ although algorithm converges for say $x_0 = 1 \in W$.
(L_3) There are no commutable error bounds on the distances $\|x_j - \zeta\|$. That is, we do not know a priori how many iterates are needed to reach a certain error tolerance.
(L_4) There are no uniqueness of the solution results.
(L_5) The more interesting semi-local convergence is not considered in Section 2.
(L_6) There is no radius of convergence. Thus, we do not know how to pick the initial point x_0.

We positively address the limitations as follows:

(L_1)′ The convergence is given for Banach space valued equations.
(L_2)′ The convergence conditions involve only the operators on the algorithm (3.1), i.e., F, F' and F''.
(L_3)′ Computable error bounds on $\|x_j - \zeta\|$ are provided. Hence, we know in advance the number of iterations to be executed to obtain a certain accuracy.
(L_4)′ A certain region is specified containing only one solution of the equation $Q(x) = 0$.
(L_5)′ The semi local convergence analysis of algorithm (2.7) is developed using majorizing sequences.
(L_6)′ The radius of convergence is specified. This is how we extend the applicability of the algorithm (2.7).

The algorithm (2.7) in a Banach space setting is given for starter $x_0 \in D$, and each $j = 0, 1, 2, \ldots$ by

$$\begin{aligned} y_j &= x_j - Q'(x_j)^{-1} Q(x_j), \\ z_j &= y_j - Q'(y_j)^{-1} Q(y_j), \\ A_j &= Q'(y_j)^2 - \frac{1}{2} Q''(y_j) Q(y_j), \\ x_{j+1} &= z_j - \frac{1}{2} A_j^{-1} Q'(y_j) A_j^{-1} Q''(y_j) Q(y_j) Q(y_j), \end{aligned} \tag{3.1}$$

where the operators Q' and Q'' denote the first and the second Fréchet derivative of Q, respectively, to solve the equation $Q(x) = 0$, where $Q : D \subset W_1 \to W_2$ is a twice Fréchet differentiable operator, D is open and convex set and W_1, W_2 are Banach spaces.
Notice that Q' is linear and Q'' is a bilinear operator [19].
It is worth noting that (2.9) cannot be written in Banach space, since the long denominator is not a linear operator.

3.1. Local analysis

The local convergence analysis is based on some conditions.
Suppose:

(C_1) There exists a function $\kappa_0 : [0, +\infty) \to \mathbb{R}$ which is continuous and also non-decreasing (CFND) such that the equation $\kappa_0(t) - 1 = 0$ has a smallest solution which is positive (SSP). Denote such solution by R_0. Set $T_0 = [0, R_0)$.

(C_2) There exists CFND $\kappa : T_0 \to \mathbb{R}$ such that for $h_1 : T_0 \to \mathbb{R}$ defined by

$$h_1(t) = \frac{\int_0^1 \kappa((1-\theta)t)d\theta}{1 - \kappa_0(t)},$$

the equation $h_1(t) - 1 = 0$ has a SSP. Denote such solution by t_1.

(C_3) The equation $\kappa_0(h_1(t)) - 1 = 0$ has a SSP in the interval T_0. Denote such solution by R_1. Set $T_1 = [0, R_1)$.

(C_4) For $h_2 : T_1 \to \mathbb{R}$ defined by

$$h_2(t) = \frac{h_1(t)\int_0^1 \kappa((1-\theta)h_1(t)t)d\theta}{1 - \kappa_0(th_1(t))},$$

the equation $h_2(t) - 1 = 0$ has a SSP. Denote such solution by t_2.

(C_5) Let $\alpha, \beta > 0$ be given constants, and $\kappa_2 : T_1 \to \mathbb{R}$ be an CFND. For $p : T_1 \to \mathbb{R}$ defined by

$$p(t) = (\kappa_0^2(th_1(t)) + 2\alpha\kappa_0\beta(th_1(t)) + \frac{\beta}{2}\kappa_2(th_1(t)))\left(\alpha + \frac{1}{\beta}\int_0^1 \kappa_0(\theta th_1(t))d\theta\right)h_1(t),$$

the equation $p(t) - 1 = 0$ has SSP. Denote such solution by R_2.
Set $T_2 = [0, R_2)$. Consider function $q : T_2 \to \mathbb{R}$ defined by

$$q(t) = \frac{\beta^2}{1 - p(t)}.$$

(C_6) The equation $\kappa_0(th_2(t)) - 1 = 0$ has an SSP. Denote such solution by R_2. Set $T_3 = [0, R_3)$.

(C_7) Let $\kappa_1 : T_3 \to \mathbb{R}$ and define functions $\bar{\kappa} : T_3 \to \mathbb{R}$ and $h_3 : T_3 \to \mathbb{R}$

$$\bar{\kappa}(t) = \begin{cases} \kappa_1(t) \\ or \\ \alpha + \kappa_0(t) \end{cases}$$

and

$$h_3(t) = \frac{h_2(t)\int_0^1 \kappa((1-\theta)th_2(t))d\theta}{1 - \kappa_0(th_2(t))} + \frac{1}{2\beta^2}q^2(t)\bar{\kappa}(th_1(t))\kappa_2(th_1(t))\left(\alpha + \frac{1}{\beta}\int_0^1 \kappa_0(\theta th_1(t))d\theta\right)^2 th_1^2(t).$$

The equation $h_3(t) - 1 = 0$ has an SSP. Denote such solution by t_3.
Set

$$t_0 = \min\{t_m\}, \quad m = 1, 2, 3$$

and $T = [0, t_0)$. Then it follows by these conditions that for each $t \in T$

$$0 \leq \kappa_0(t) < 1, \tag{3.2}$$

$$0 \leq \kappa_0(t)th_1(t) < 1, \tag{3.3}$$

$$0 \leq \kappa_0(t)th_2(t) < 1, \tag{3.4}$$

$$0 \leq p(t) < 1, \tag{3.5}$$

$$0 \leq q(t), \tag{3.6}$$

and

$$0 \leq h_m(t) < 1. \tag{3.7}$$

The constant t_0 is proven to be a radius of convergence for the algorithm (3.1) in Theorem 1. First, we relate the developed functions $\kappa, \kappa_0, \kappa_1$ and κ_2 to the operators on the algorithm (3.1).

(C_8) There exists a solution $\zeta \in D$ of the Eq (3.1), an invertible linear operator M such that $\|M\| \leq \alpha$ and $\|M^{-1}\| \leq \beta$.

(C_9) $\|Q'(x) - M\| \leq \frac{1}{\beta}\kappa_0(\|x - \zeta\|)$ for each $x \in D$. Define the region $D_0 = D \cap S(\zeta, R_0)$, where $S(\zeta, R_0)$ stands for a ball that is open having center ζ and radius $R_0 > 0$. It follows by this condition and (C_1) that for $x = \zeta$

$$\|M^{-1}\| \, \|Q'(x) - M\| \leq \kappa_0\|x - \zeta\| < 1.$$

Thus, from the standard lemma by Banach on linear, and invertible operators [20, 21] $Q'(x)$ is invertible, and

$$\|Q'(x)^{-1}\| \leq \frac{\beta}{1 - \kappa_0(\|x - \zeta)\|)}. \tag{3.8}$$

(C_{10}) $\|Q'(x) - Q'(y)\| \leq \frac{1}{\beta}\kappa(\|x - y\|)$ for each $x, y \in D_0$.
Set $D_1 = D \cap S(\zeta, R_3)$.

(C_{11}) $\|Q'(x)\| \leq \frac{1}{\beta}\kappa_1(\|x - \zeta\|)$ and $\|Q''(x)\| \leq \frac{1}{\beta}\kappa_2(\|x - \zeta\|)$ for each $x \in D_1$.

(C_{12}) $S[\zeta, t_0] \subset D$, where $S[\zeta, t_0]$ stands for the closure of $S(\zeta, t_0)$.
Notice that the first condition in (C_{11}) can be dropped, since

$$\|Q'(x)\| \leq \|Q'(x) - M + M\| \leq \|M\| + \|Q'(x) - M\| \leq \alpha + \frac{1}{\beta}\kappa_0(\|x - \zeta\|).$$

Hence, the function κ_1 can be defined by $\kappa_1(t) = \alpha + \frac{1}{\beta}\kappa_0(t)$. Moreover, the function $\bar{\kappa}$ given in the condition (C_7) is chosen in particular to be the smallest of the functions κ_1 and $\alpha + \frac{1}{\beta}\kappa_0(t)$. The developed notation and the conditions (C_1)–(C_{12}) shall be used in the local analysis of convergence for the algorithm (3.1).

Theorem 3. *Suppose that the conditions (C_1)–(C_{12}) are satisfied, and $x_0 \in S_0 = S(\zeta, t_0) \setminus \{\zeta\}$. Then, the following assertions hold for each $j = 0, 1, 2, \ldots$*

$$\{x_j\} \subset S(\zeta, t_0), \tag{3.9}$$

$$\|y_j - \zeta\| \leq h_1(\|x_j - \zeta\|)\|x_j - \zeta\| \leq \|x_j - \zeta\| < t_0, \tag{3.10}$$

$$\|z_j - \zeta\| \leq h_2(\|x_j - \zeta\|)\|x_j - \zeta\| \leq \|x_j - \zeta\|, \tag{3.11}$$

$$\|x_{j+1} - \zeta\| \leq h_3(\|x_j - \zeta\|)\|x_j - \zeta\| \leq \|x_j - \zeta\|, \tag{3.12}$$

and the sequence $\{x_j\}$ *is convergent to* ζ.

Proof. By assuming $x_0 \in S_0$, the assertion (3.9) is satisfied for $j = 0$. Let $x \in S(\zeta, t_0)$. Then, by the condition (C_1), $Q'(x)$ is invertible, and the estimate (3.8) holds. In particular, for $x = x_0$ the linear operator $Q'(x_0)$ is invertible, since $x_0 \in S_0$. Thus, the iterate y_0 is well defined. Then, we can write by the first substep of the algorithm (3.1) for $j = 0$:

$$y_0 - \zeta = x_0 - \zeta - Q'(x_0)^{-1}Q(x_0) = Q'(x_0)^{-1}[Q'(\zeta + \theta(x_0 - \zeta)) - Q'(x_0)]d\theta(x_0 - \zeta). \tag{3.13}$$

By (3.8), the condition (3.9), the definition of the function h_1, (3.7), and (3.13), we have in turn

$$\begin{aligned} \|y_0 - \zeta\| &\leq \frac{\beta}{1 - \kappa_0(\|x_0 - \zeta\|)} \cdot \frac{\int_0^1 \kappa((1-\theta)\|x_0 - \zeta\|)d\theta}{\beta}\|x_0 - \zeta\| \\ &\leq h_1(\|x_0 - \zeta\|)\|x_0 - \zeta\| \leq \|x_0 - \zeta\| < t_0, \end{aligned} \tag{3.14}$$

so the assertion (3.10) is satisfied, the iterate $y_0 \in S(\zeta, t_0)$, and consequently (3.8) holds if $x = y_0$. Thus, the iterate z_0 is well defined. By the second substep of the algorithm (3.1) for $j = 0$:

$$z_0 - \zeta = y_0 - \zeta - Q'(y_0)^{-1}Q(y_0) = Q'(y_0)^{-1}[Q'(\zeta + \theta(y_0 - \zeta)) - Q'(y_0)]d\theta(y_0 - \zeta), \tag{3.15}$$

leading as in (3.13) and (3.14) to

$$\|z_0 - \zeta\| \leq h_2(\|x_0 - \zeta\|)\|x_0 - \zeta\| \leq \|x_0 - \zeta\|, \tag{3.16}$$

showing the assertion (3.11), the iterate $z_0 \in S(\zeta, t_0)$, and the estimate (3.8) holds for $x = z_0$. We shall show the invertability of the operator A_0, which implies the existence of the iterate x_1. Consider the identity

$$\begin{aligned} A_0 - M^2 &= (Q'(y_0) - M + M)^2 - \frac{1}{2}Q''(y_0)Q(y_0) - M^2 \\ &= (Q'(y_0) - M)^2 + (Q'(y_0) - M)M + M(Q'(y_0) - M) + M^2 - \frac{1}{2}Q''(y_0)Q(y_0) - M^2 \\ &= (Q'(y_0) - M)^2 + (Q'(y_0) - M)M + M('Q(y_0) - M) - \frac{1}{2}Q''(y_0)Q(y_0). \end{aligned} \tag{3.17}$$

In view of (C_7), the second condition in (C_{11}), (C_5), (3.5)–(3.7), and (3.17), we get

$$\begin{aligned} \|M^{-2}\|\,\|A_0 - M^2\| \leq \beta^2\Big[&\frac{\kappa_0^2(\|y_0 - \zeta\|)}{\beta^2} + 2\frac{\alpha}{\beta}\kappa_0(\|y_0 - \zeta\|) + \frac{1}{2}\frac{\kappa_2(\|y_0 - \zeta\|)}{\beta} \\ &(\alpha + \frac{1}{\beta}\int_0^1 \kappa_0(\theta\|y_0 - \zeta\|)d\theta)\|y_0 - \zeta\|\Big] = p_0 < 1, \end{aligned} \tag{3.18}$$

where we also use $S_0\|A_0^{-1}\| \leq q_0$, (3.8)

$$\|Q(y_0)\| = Q(y_0) - Q(\zeta) = \int_0^1 Q'(\zeta + \theta(y_0 - \zeta))d\theta\|y_0 - \zeta\|. \tag{3.19}$$

Thus,

$$\begin{aligned}\|Q(y_0)\| &\leq \| \int_0^1 Q'(\zeta + \theta(y_0 - \zeta))d\theta - N + M\| \\ &\leq \alpha + \frac{1}{\beta}\int_0^1 \kappa_0(\theta\|y_0 - \zeta\|)d\theta.\end{aligned}$$

Then, we can write by the third substep of algorithm (3.1)

$$x_1 - \zeta = z_0 - \zeta - \frac{1}{2}A_0^{-1}Q'(y_0)A_0^{-1}Q''(y_0)Q(y_0)Q(y_0). \tag{3.20}$$

It follows by (C_8)–(C_{11}), (3.7) (for $j = 3$), (3.14), (3.15), (3.19), and (3.20) that

$$\begin{aligned}\|x_1 - \zeta\| &\leq \|z_0 - \zeta| + \frac{1}{2}\|A_0^{-1}\|^2\|Q'(y_0)\|\,\|Q''(y_0)\|\,\|Q(y_0)\|^2 \\ &\leq \frac{\int_0^1 \kappa((1-\theta)\|z_0 - \zeta\|)d\theta\|z_0 - \zeta\|}{1 - \kappa_0(\|z_0 - \zeta\|)} + \frac{1}{2}q_0^2\frac{\bar{\kappa}}{\beta^2}(\|y_0 - \zeta\|)\kappa_2(\|y_0 - \zeta\|)(\alpha \\ &+ \frac{1}{\beta}\int_0^1 \kappa_0(\theta\|y_0 - \zeta\|)d\theta)^2\|y_0 - \zeta\|^2 \\ &\leq h_3(\|x_0 - \zeta\|)\|x_0 - \zeta\| \leq \|x_0 - \zeta\|,\end{aligned} \tag{3.21}$$

showing the assertion (3.12) and $x_1 \in S(\zeta, t_0)$. Simply switch x_0, y_0, z_0, x_1 by x_k, y_k, z_k, x_{k+1} in the preceding calculations to complete the induction for the assertions (3.9)–(3.12). Then, from the estimation

$$\|x_{k+1} - \zeta\| \leq \|x_k - \zeta\| < t_0,$$

we deduce that $x_{k+1} \in S(\zeta, t_0)$ and

$$\lim_{k\to+\infty} x_k = \zeta.$$

□

Next, a neighborhood of ζ is determined that contains no other solution.

Proposition 1. *Suppose there exists a solution $\zeta \in S(\zeta, t_4) \subset D$ of the equation $Q(x) = 0$, such that the condition (C_9) holds on $S(\zeta, t_4)$, and there exists $t_5 \geq t_4$ so that*

$$\frac{1}{\beta}\int_0^1 \kappa_0(\theta t_5)d\theta < 1. \tag{3.22}$$

Define the region $D_2 = D \cap S[\zeta, t_5]$. Then, the only solution of the Eq (1.1) *in the region D_2 is ζ.*

Proof. Let $E = \int_0^1 Q'(\zeta + \theta(u - \zeta))d\theta$ for some $u \in D_2$ with $Q(u) = 0$. By applying the condition (C_9), and the assumption (3.22), we obtain in turn

$$\|M^{-1}\|\,\|E - M\| \leq \int_0^1 \kappa_0(\theta(u - \zeta))d\theta \leq \int_0^1 \kappa_0(\theta t_5)d\theta < 1,$$

so E is invertible. Then, from the identity

$$u - \zeta = E^{-1}(Q(u) - Q(\zeta)) = E^{-1}(0) = 0,$$

we deduce that $u = \zeta$. □

3.2. Semi-local analysis

This analysis is based on majorizing sequences. The role of x_0 and the κ function is replaced by ζ and the ϑ.

Suppose:

(H_1) There exists CFND $\vartheta_0 : [0, +\infty) \to \mathbb{R}$ such that equ ation $\vartheta_0(t) - 1 = 0$ has an SSP. Denote such solution by δ_0.

(H_2) There exist CFND $\vartheta, \vartheta_1, \vartheta_2 : [0, \delta_0) \to \mathbb{R}$. Define the sequences $\{a_j\}, \{b_j\}, \{c_j\}$ for $a_0 = 0$, some $b_0 \geq 0$, and each $j = 0, 1, \ldots$, by

$$\begin{aligned}
s_j &= \frac{1}{\beta}\int_0^1 v((1-\theta))d\theta(b_j - a_j),\\
c_j &= b_j + \frac{s_j}{1-\vartheta_0(b_j)}\\
\lambda_j &= \vartheta_0^2(b_j) + 2\alpha\beta\vartheta_0(b_j) + \frac{\beta}{2}\vartheta_2(b_j)s_j,\\
\mu_j &= \frac{\beta^2}{1-\lambda_j},\\
\bar{\vartheta} &= \begin{cases} \vartheta_1(b_j) \\ \text{or} \\ \alpha + \dfrac{\vartheta_0(b_j)}{\beta}, \end{cases}\\
a_{j+1} &= c_j + \frac{1}{2}\mu_n^2\frac{\vartheta(\bar{b}_j)}{\beta}\frac{\vartheta_2(b_j)}{\beta}s_j^2,\\
\beta\gamma_{j+1} &= \int_0^1 \vartheta((1-\theta)(a_{j+1}-a_j))d\theta(a_{j+1}-a_j)\\
&\quad + (\alpha + \frac{1}{\beta}\vartheta_0(a_j))(a_{j+1} - b_j),\\
b_{j+1} &= a_{j+1} + \frac{\gamma_{j+1}}{1-\vartheta_0(a_{j+1})}.
\end{aligned} \tag{3.23}$$

These scalar sequences are shown to be majorizing for the algorithm (3.1) in Theorem 2. However, a convergence condition for these sequences is needed.

(H_3) There exists $\delta \in [0, \delta_0)$ such that for each $j = 0, 1, 2, \ldots$

$$\vartheta_0(a_j) < 1, \vartheta_0(b_j) < 1, \lambda_j < 1, \text{ and } a_j \geq \delta.$$

It follows by this definition and (3.23) that $0 \leq a_j \leq b_j \leq c_j \leq a_{j+1} \leq \delta$, and there exists $a^* \in [0, \delta]$ such that

$$\lim_{j \to +\infty} a_j = a^*.$$

The functions ϑ relate to the operator on the algorithm.

(H_4) There exist constants $\alpha, \beta > 0$, and an invertiable operator M.

(H_5) For some $x_0 \in D$, and each $x \in D$

$$\|Q'(x) - M\| \leq \frac{1}{\beta}\vartheta_0(\|x - x_0\|).$$

Set $D_3 = D \cap S(x_0, \delta_0)$.

Notice that if $x = x_0$, we get $\|M^{-1}\| \|Q'(x_0) - M\| \leq \vartheta_0(0) < 1$, by the definition of δ_0. Thus, the linear operator $Q'(x_0)$ is invertible. Set $\|Q'(x_0)^{-1}Q(x_0)\| \leq b_0$.

(H_6) For each $x, y \in D_3$

$$\|Q'(x) - Q'(y)\| \leq \frac{1}{\beta}\vartheta(\|x - y\|),$$
$$\|Q'(x)\| \leq \frac{1}{\beta}\vartheta_1(\|x - x_0\|),$$
$$\|Q''(x)\| \leq \frac{1}{\beta}\vartheta_2(\|x - x_0\|).$$

(H_7) $S[x_0, a^*] \subset D$.

Next, the semilocal analysis of convergence for the algorithm (3.1) is developed under the conditions (H_1)–(H_7).

Theorem 4. *Suppose that the conditions (H_1)–(H_7) hold. Then, the sequence $\{x_j\}$ generated by the algorithm* (3.1)*, exists in $S(x_0, a^*)$, stays in $S(x_0, a^*)$ for each $j = 0, 1, 2, \ldots$, and is convergent to a solution $\zeta \in S[x_0, a^*]$ of the equation $Q(x) = 0$, such that for each $j = 0, 1, 2, \ldots$*

$$\|\zeta - x_j\| \leq a^* - a_j. \tag{3.24}$$

Proof. Mathematical induction is utilized to show the assertions

$$\|y_j - x_j\| \leq b_j - a_j, \tag{3.25}$$
$$\|z_j - y_j\| \leq c_j - b_j, \tag{3.26}$$
$$\|x_{j+1} - z_j\| \leq a_{j+1} - c_j. \tag{3.27}$$

The assertion (3.26) holds if $j = 0$, since by the first substep of algorithm (3.1), and the definition of b_0:

$$\|y_0 - x_0\| = \|Q'(x_0)^{-1}Q(x_0)\| \leq b_0 = b_0 - a_0 < a^*.$$

Thus, the iterate $y_0 \in S(x_0, a^*)$. We can write by the first substep of (3.1) that

$$Q(y_0) = Q(y_0) - Q(x_0) - Q'(x_0)(y_0 - x_0) = [Q'(x_0 + \theta(y_0 - x_0)) - Q'(x_0)]d\theta(y_0 - x_0). \tag{3.28}$$

Using the first condition in (H_6)

$$\begin{aligned} \|Q(y_0)\| &\leq \frac{1}{\beta}\int_0^1 \vartheta((1-\theta)\|y_0 - x_0\|)d\theta\|y_0 - x_0\| \\ &\leq \frac{1}{\beta}\int_0^1 \vartheta((1-\theta)(b_0 - a_0))d\theta(b_0 - a_0) = \delta_0. \end{aligned} \tag{3.29}$$

By the condition (H_4) for $x = y_0$

$$\|M^{-1}\|\,\|Q'(y_0) - M\| \leq \vartheta_0(\|y_0 - x_0\|) \leq \vartheta_0(b_0) < 1,$$

Thus, $Q(y_0)$ is invertible, and

$$\|Q'(y_0)^{-1}\| \leq \frac{\beta}{1 - \kappa_0(b_0)}, \tag{3.30}$$

and the iterate z_0 is well defined by the second substep of (3.1). Then, we can also write

$$z_0 - y_0 = Q'(y_0)^{-1}Q(y_0). \tag{3.31}$$

In view of (3.23) and (3.29)–(3.31), we get

$$\|z_0 - y_0\| \leq \frac{\delta_0}{1 - \vartheta_0(b_j)} = c_0 - b_0,$$

and

$$\|z_0 - x_0\| \leq \|z_0 - y_0\| + \|y_0 - x_0\| \leq c_0 - b_0 + b_0 - a_0 = c_0 < a^*.$$

Hence, the assertion (3.29) holds if $j = 0$, and $z_0 \in S(x_0, a^*)$.

As in the local case:

$$\begin{aligned} \|M^{-2}\|\,\|A_0 - M^2\| &\leq \beta^2\Big[\frac{\vartheta_0^2(\|y_j - x_0\|)}{\beta^2} + 2\frac{\alpha}{\beta}\vartheta_0(\|y_j - x_0\|) + \frac{1}{2\beta}\vartheta_2(\|y_j - x_0\|)\delta_j\Big] \\ &\leq \beta^2\Big[\frac{\vartheta_0^2(b_j)}{\beta^2} + \frac{2\alpha}{\beta}\vartheta_0(b_j) + \frac{1}{2\beta}\vartheta_0(b_j) + \frac{1}{2\beta}\vartheta_2(b_j)\delta_j\Big] \\ &= \lambda_j < 1, \end{aligned}$$

so, A_0 is invertible, and

$$\|A_0^{-1}\| \leq \mu_0. \tag{3.32}$$

Notice, also that $Q'(z_0)$ is invertible, so the iterate x_1 is well defined by the third substep of (3.1). Then, we have by (3.1)

$$x_1 - z_0 = -\frac{1}{2}A_0^{-1}Q'(y_0)A_0^{-1}Q''(y_0)Q(y_0)Q(y_0),$$

leading to

$$\|x_1 - z_0\| \leq \frac{1}{2}\mu_0^2 \frac{\bar{\vartheta}(b_0)}{\beta}\frac{\vartheta_2(b_0)}{\beta}\delta_0^2 = a_1 - c_0,$$

and

$$\|x_1 - z_0\| \leq \|x_1 - z_0\| \leq +\|z_0 - x_0\| \leq a_1 - c_0 + c_0 - a_0 = a_1 < a^*.$$

Thus, the assertion (3.30) holds for $j = 0$, and the iterate $x_1 \in S(z_0, a^*)$. We can write

$$Q(x_{m+1}) = Q(x_{m+1}) - Q(x_m) - Q'(x_m)(x_{m+1} - x_m) + Q'(x_m)(y_m - x_m),$$

leading to

$$\begin{aligned}\|Q(x_{m+1})\| &\leq \frac{1}{\beta}\int_0^1 \vartheta((1-\theta)\|x_{m+1} - x_m\|)d\theta\|x_{m+1} - x_m\| + (\alpha + \frac{1}{\beta}\vartheta_0(\|x_m - x_0\|))\|x_{m+1} - y_m\| \\ &\leq \frac{1}{\beta}\int_0^1 \vartheta((1-\theta)(a_{m+1} - a_m))d\theta(a_{m+1} - a_m) + (\alpha + \frac{1}{\beta}\vartheta_0(a_m))(a_{m+1} - b_m) = \gamma_{m+1},\end{aligned} \tag{3.33}$$

so

$$\begin{aligned}\|y_{m+1} - x_{m+1}\| &\leq \|Q'(x_{m+1})^{-1}\|\,\|Q(x_{m+1})\| \\ &\leq \frac{\gamma_{m+1}}{1 - \vartheta_0(\|x_{m+1} - x_0\|)} \leq \frac{\gamma_{m+1}}{1 - \vartheta_0(a_{m+1})} \\ &= b_{m+1} - a_{m+1},\end{aligned} \tag{3.34}$$

and

$$\|y_{m+1} - x_0\| \leq \|y_{m+1} - x_{m+1}\| + \|x_{m+1} - x_0\| \leq b_{m+1} - a_{m+1} + a_{m+1} - a_0 = b_{m+1} < a^*.$$

The induction for the assertions (3.25)–(3.27) is completed if x_0, y_0, z_0, x_1 are replaced by x_m, y_m, z_m, x_{m+1}, respectively. Then, we have

$$\|x_{m+1} - x_m\| \leq \|x_{m+1} - y_m\| + \|y_m - x_m\| \leq a_{m+1} - b_m + b_m - a_m = a_{m+1} - a_m.$$

Consequently, the sequence $\{x_m\}$ is Cauchy in a Banach space and such it is convergent to some $\zeta \in S[x_0, a^*]$. By letting $m \to +\infty$ in (3.33) we get $Q(\zeta) = 0$. Moreover, by the estimate

$$\|x_{j+k} - x_j\| \leq a_{j+k} - a_j, \tag{3.35}$$

we obtain (3.24) by letting $k \to +\infty$ in (3.34). □

As in the local case a neighborhood of x_0 is special containing only one solution in the next result.

Proposition 2. *Suppose there exists a solution $y^* \in S(x_0, t_7)$ of the Eq* (3.1).

The condition (H_5) holds in ball $S(x_0, t_7)$, and there exists $t_8 \geq t_7$ such that

$$\beta\int_0^1 \vartheta_0(\theta t_7 + (1-\theta)t_8)d\theta < 1. \tag{3.36}$$

Define the region $D_4 = D \cap S[x_0, t_8]$.

Then, the only solution of the equation $Q(x) = 0$ in the region D_4 is y^*.

Proof. Let $E_1 = \int_0^1 Q'(y^*+\theta(z^*-y^*)d\theta$ for some $z^* \in D_4$ with $Q(z^*) = 0$. It follows by the condition (H_5) and (3.36)

$$\|M^{-1}\| \, \|E_1 - M\| \leq \beta \int_0^1 \vartheta_0(\theta\|y^* - x_0\| + (1-\theta)\|z^* - x_0\|)d\theta \leq \beta \int_0^1 \vartheta_0(\theta t_7 + (1-\theta)t_8)d\theta < 1.$$

Therefore, the operator E_1 is invertible. Then, from the identity

$$z^* - y^* = E_1^{-1}(Q(z^*) - Q(y^*)) = E^{-1}(0) = 0,$$

we deduce that $z^* = y^*$. □

Remark 1. *(i) Popular choices for $M = Q'(\zeta)$ (local case) and $M = Q'(x_0)$ (semilocal case). However, these choices are not the necessary the most flexible. Other choices exist [22, 23].*

(ii) Note that not all the conditions (H_1)–(H_7) are assumed in Proposition 1. However, if they are, we can take $y^* = \zeta$, and $t_7 = a^*$.

(iii) The constant δ_0 can replace the limit point a^* is the condition (H_7).

4. Visual analysis via polynomiography

The analysis of stability and speed of convergence is an essential aspect of analyzing every root-finding algorithm. The standard way of performing such an analysis is the use of visual approach via the so-called polynomiography [24, 25]. In polynomiography, we generate images called polynomiographs using a general algorithm presented in Algorithm 1. Depending on the coloring algorithm in the last step of the algorithm, we can visualize various aspects of the root-finding algorithm.

Algorithm 1: Generation of a polynomiograph.

Input: $p \in \mathbb{C}[Z]$, $\deg p \geq 2$ – polynomial; R – root finding algorithm; $A \subset \mathbb{C}$ – area; N – the maximum number of iterations; ε – accuracy; *colors* – color map.

Output: Polynomiograph for the complex-valued polynomial p within the area A.

```
for z_0 ∈ A do
    n = 0
    while n < N do
        z_{n+1} = R(z_n, p)
        if |p(z_{n+1})| < ε then
            break
        n = n + 1
    Determine the color for z_0 and assign the color using color map colors
```

In this section, we use a coloring algorithm that shows basins of attraction and speed of convergence in the same polynomiograph [26]. In this algorithm, each root of the considered polynomial gets a distinct color. We also need an additional color to depict the non-convergent points (we use the black

color for this aim). Now, to color the given starting point z_0, we first check whether the root-finding algorithm has converged to any of the roots, i.e., if $n < N$, where n is the number of the last iterate. If the algorithm has converged, then, for the found root, we find the closest root of the polynomial and use its color to color z_0, else we use the additional color. In this way, we show basins of attraction. To add the information on the speed of convergence, we change the shade of the color based on the number of performed iteration. In this way, a light color will show a small number of iterations and a dark color a large number of iterations.

Based on polynomiographs, we can compute some numerical measures that will allow the analysis to be broadened. The three most widely used measures are: The average number of iterations (ANI) [27], convergence area index (CAI) [27], and the computations time [28]. To compute the average number of iterations, we take the information on the speed of convergence in the polynomiograph because it contains the number of iterations needed to find the root for all points in the considered area A. CAI is the ratio of the number of starting points that converged to any root to the number of all points in the considered area A. The CAI value is a real number between 0 and 1, and it gives us information on the percentage of points in A that have converged to roots.

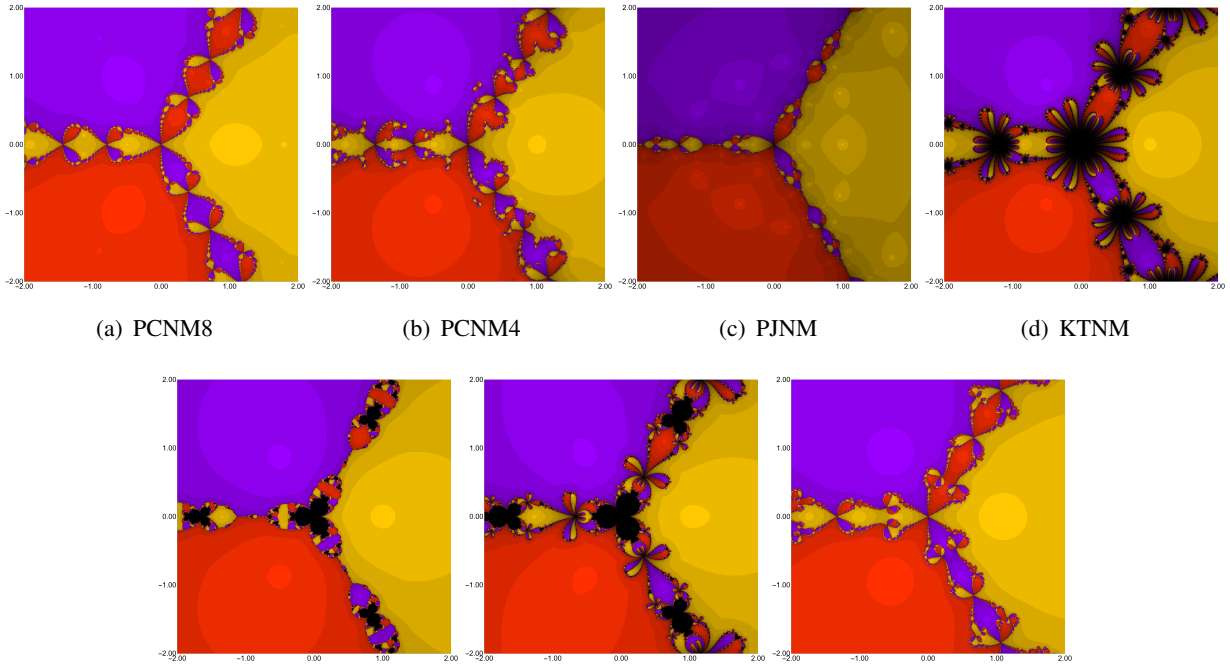
The polynomiographs used for the analysis presented in this section were generated for the following common parameters: $N = 30$, $\varepsilon = 0.001$, and polynomiograph resolution 1000×1000 pixels. In our analysis, we use three polynomials:

- $p_3(z) = z^3 - 1$, roots: $1, -0.5 + \frac{\sqrt{3}}{2}i, -0.5 - \frac{\sqrt{3}}{2}i$, the area $A = [-2, 2]^2$,
- $p_4(z) = z^4 - 10z^2 + 9$, roots: $-3, -1, 1, 3$, the area $A = [-4, 4]^2$,
- $p_5(z) = z^5 - z$, roots: $0, -1, 1, -i, i$, the area $A = [-2, 2]^2$.

The program for generating polynomiographs was implemented in Mathematica 13.2 using the parallelization option of the `Compile` command. Moreover, the polynomiographs were generated on the computer with the specifications: Intel i5-9600K (@3.70 GHz), 32 GB DDR4 RAM, and Windows 10 (64 bit). Apart from the two proposed algorithms, PCNM8 and PCNM4, we generated polynomiographs for the PJNM, KTNM, CNM1, CNM2, and ONM algorithms for comparison purposes.

The obtained results for the p_3 polynomial are presented in Figure 2 (the polynomiographs) and in Table 1 (the numerical measures). From the polynomiographs, we see that each algorithm has different basins of attraction. However, in each case, we see three characteristic braids. The degree of interweaving of the basins tells us about the stability of the algorithm (the less the interweaving, the better the stability of the algorithm). Based on the polynomiographs, we see that the best stability is obtained by the PJNM algorithm, followed by the PCNM8 and ONM algorithms. For the CNM1 algorithm, we also see a low degree of interweaving, but there are also non-convergent points (the black areas) which are not visible in the three algorithms mentioned. This is confirmed by the CAI measure presented in Table 1. The PJNM, PCNM8 and ONM algorithms all obtained values equal to 1.0, whereas for the CNM1 algorithm, the CAI value is less than 1.0, namely it is equal to 0.967. The lowest value of CAI was obtained by the CNM2 algorithm (0.946). When it comes to the speed of convergence, we see that the lightest colors are visible in the polynomiographs for the PCNM8 and ONM algorithms, followed by the PCNM4 and CNM1 algorithms. Again, we can confirm this observation by looking at the values of ANI gathered in Table 1. The lowest value of ANI was obtained by the PCNM8 algorithm (1.496), and the second lowest value equal to 1.624 was obtained by the ONM

algorithm. Now, if we look at the generation time of polynomiographs, we see that the fastest algorithm is the CNM1 algorithm (0.762 s). The two best algorithms in terms of ANI, i.e., the PCNM8 and ONM, obtained slightly slower times equal to 0.778 and 0.775 s, respectively. The slowest algorithm is the KTNM algorithm (0.914 s).



(a) PCNM8 (b) PCNM4 (c) PJNM (d) KTNM

(e) CNM1 (f) CNM2 (g) ONM

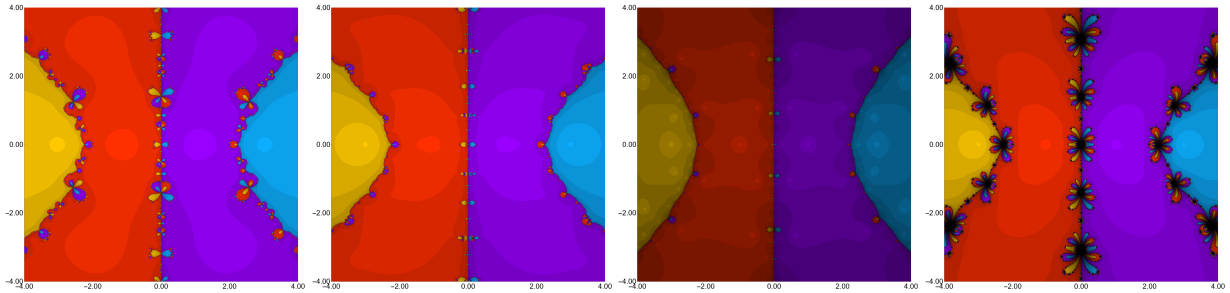
Figure 2. Polynomiographs for the complex polynomial p_3 generated using various root-finding algorithms.

Table 1. Numerical measures obtained from the polynomiographs generated for p_3 (Figure 2).

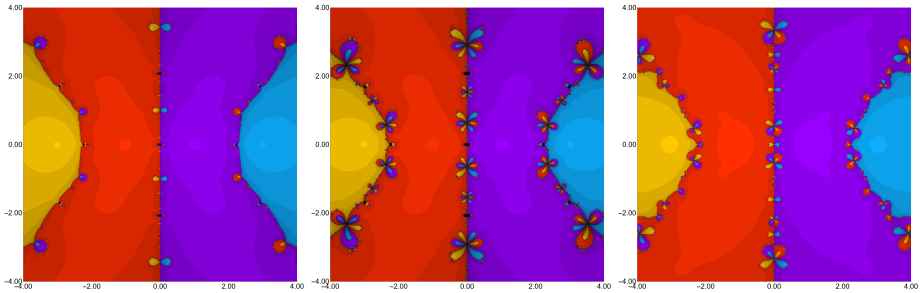
Algorithm	ANI	CAI	Time [s]
PCNM8	1.496	1.0	0.778
PCNM4	2.320	1.0	0.774
PJNM	5.562	1.0	0.832
KTNM	5.358	0.973	0.914
CNM1	2.642	0.967	0.762
CNM2	3.840	0.946	0.808
ONM	1.624	1.0	0.775

In the next example, we present and analyze the results obtained for the p_4 polynomial. In Figure 3, we see the polynomiographs for this polynomial, whereas in Table 2, we gather the values of numerical measures. In the polynomiographs, we see four basins of attraction. The most significant difference in those basins is visible at the boundaries, where the interweaving of the basins appears. The lowest degree of interweaving can be observed for the PCNM4, PJNM, and CNM1 algorithms. Thus, these

three algorithms obtained the best stability among the analyzed algorithms. However, for the CNM1 algorithm, we can also observe some non-convergent points in the polynomiograph, which causes the CAI value to be equal to 0.999. The other two algorithms obtained a CAI value of 1.0. There are also two other algorithms with full convergence (CAI value equal to 1.0), namely the PCNM8 and ONM algorithms. In terms of speed of convergence, the best results are obtained by the ONM and PCNM8 algorithms, for which the ANI value is equal to 1.566 and 1.660, respectively. We can observe this in the polynomiographs because these two algorithms have the biggest light areas. On the other extreme, the worst speed of convergence can be observed for the PJNM algorithm (the darkest colors and the value of ANI equal to 7.561). If we look at the generation times, which reflect the real time of computations, we see that the fastest algorithm was the PCNM8 algorithm, followed by the CNM1 algorithm. These two algorithms obtained the times equal to 0.781 and 0.840 s, respectively. The slowest algorithm among the analyzed algorithms was the PJNM algorithm, which obtained a time equal to 1.085 s.



(a) PCNM8 (b) PCNM4 (c) PJNM (d) KTNM



(e) CNM1 (f) CNM2 (g) ONM

Figure 3. Polynomiographs for the complex polynomial p_4 generated using various root-finding algorithms.

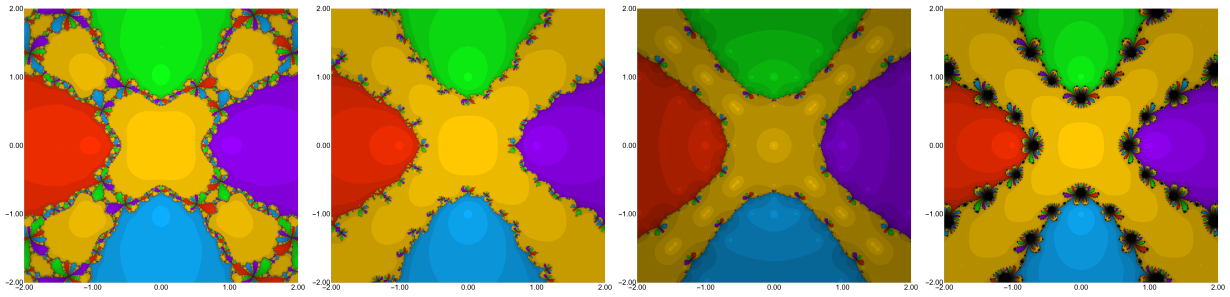
Table 2. Numerical measures obtained from the polynomiographs generated for p_4 (Figure 3).

Algorithm	ANI	CAI	Time [s]
PCNM8	1.660	1.0	0.781
PCNM4	2.212	1.0	0.901
PJNM	7.561	1.0	1.085
KTNM	3.948	0.993	0.892
CNM1	2.201	0.999	0.840
CNM2	2.593	0.998	0.858
ONM	1.566	1.0	0.844

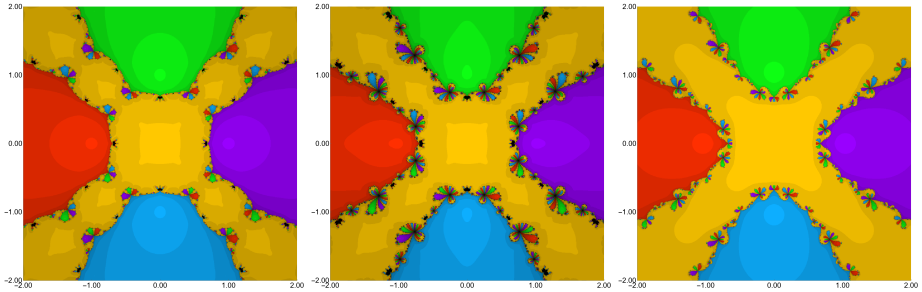
In the last example, we visually analyze the results obtained for the p_5 polynomial. The polynomiographs generated for this polynomial are presented in Figure 4, and the calculated numerical measures are in Table 3. The best stability is observed for the PJNM algorithm, followed by the CNM1 algorithm. For these two algorithms, the interweaving of the basins is the smallest. However, for the CNM1 algorithm, we can observe areas of non-convergent points, whereas, for the PJNM algorithm, we do not see any such points. The values of CAI gathered in Table 3 confirm this observation, where the PJNM and CNM1 algorithms obtained values of 1.0 and 0.997, respectively. The highest value of CAI (1.0) is also obtained by three other analyzed algorithms, namely by the PCNM8, PCNM4, and ONM algorithms. The lowest value of 0.982 is obtained by the KTNM algorithm, which is clearly visible in the polynomiograph, where we see many black areas. Now, if we analyze the speed of convergence, then we notice that the fastest algorithms are the ONM and PCNM8 algorithms, for which we see the lightest shades of the basins' colors. Consequently, these two algorithms obtained the lowest values of ANI, equal to 1.556 and 2.013, respectively. The worst speed of convergence is visible for the PJNM algorithm. Moreover, this algorithm also obtained the highest ANI value, which was equal to 5.678. When it comes to the generation times, like for the p_3 polynomial, the shortest time of 0.834 s was obtained by the CNM1 algorithm. The second best algorithm with a time equal to 0.838 s is the PCNM8 algorithm.

Table 3. Numerical measures obtained from the polynomiographs generated for p_5 (Figure 4).

Algorithm	ANI	CAI	Time [s]
PCNM8	2.013	1.0	0.838
PCNM4	2.413	1.0	0.930
PJNM	5.678	1.0	0.994
KTNM	4.378	0.982	0.931
CNM1	2.321	0.997	0.834
CNM2	2.758	0.994	0.878
ONM	1.556	1.0	0.859



(a) PCNM8 (b) PCNM4 (c) PJNM (d) KTNM



(e) CNM1 (f) CNM2 (g) ONM

Figure 4. Polynomiographs for the complex polynomial p_5 generated using various root-finding algorithms.

5. Numerical results

In this section, we investigate the effectiveness of newly proposed fourth- and eighth-order algorithms for solving several nonlinear real-world application problems including single and multi-variables. Several parameters have been specified to compare the proposed algorithms with some of the existing ones having similar order of convergence. The parameters include the number of iterations (NI), the number of function evaluations (NFE), the absolute error at the final iteration ($|\eta| = |x_N - x_{N-1}|$), the absolute value of the nonlinear function ($|\psi(x_N)|$), and the CPU time (in seconds). As the stopping criterion, we use the following tolerance:

$$|\eta| = |x_N - x_{N-1}| \leq 10^{-300}.$$

Problem 1. *For the first nonlinear model, we consider a path traversed by an electron in the air gap between two parallel plates considering the multi-factor effect, which is given by [29]*

$$\mu(x) = \mu_0 + \left(v_0 + c_0 \frac{E_0}{m\omega} \sin \omega x_0 + \sigma\right)(x - x_0) + c_0 \frac{E_0}{m\omega^2}(\cos(\omega x + \sigma) + \sin(\omega x + \sigma)), \tag{5.1}$$

where μ_0 and v_0 are the position and velocity of the electron at time x, m and c_0 are the mass and the charge of the electron at rest, respectively, and $E_0 \sin(\omega x + \sigma)$ is the radio frequency electric field between the plates.

If particular parameters are chosen, Eq (5.1) can be simplified as

$$\psi_1(x) = x - 0.5 \cos x + \frac{\pi}{4} = 0. \tag{5.2}$$

The results of numerical simulations of an engineering application Problem 1 carried out for a fixed number of iterations, that is, when NI = 7 and $x_0 = 10.5$ are presented in Table 4. It is observed that the proposed fourth-order PCNM4 algorithm produces, at the final iteration, the smallest absolute error and the smallest absolute value of the function itself when compared to the other existing algorithms. This proposed fourth-order algorithm has the advantage of less CPU time consumption in comparison to other fourth-order existing algorithms. Similarly, in Table 5, it is observed that the existing algorithm of eight-order convergence (ONM) takes the fewest number of iterations (NI) in comparison to the other algorithms, while the proposed eight-order PCNM8 algorithm produces the smallest absolute error when compared to the other algorithms. In Table 5, we observe that the existing algorithms of order four PJNM, KTNM, CNM1, and CNM2 produce the smallest absolute errors when compared to the proposed fourth-order PCNM4, while the PCNM4 algorithm takes a fewer number of iterations in comparison to the other fourth-order existing algorithms.

Table 4. Numerical simulations for Problem 1 for initial guess ($x_0 = 10.5$) and number of iterations (NI= 7).

Algorithm	$\|\eta\|$	$\|\psi_1(x_N)\|$	CPU Time [s]
PCNM4	2.5741e-505	8.2489e-2021	4.88e-01
PJNM	2.9844e-268	7.5068e-1073	9.68e-01
KTNM	1.2395e-135	2.0495e-541	9.06e-01
CNM1	8.5050e-47	5.7423e-94	7.35e-01
CNM2	8.5050e-47	5.7423e-94	5.78e-01

Table 5. Numerical simulations for Problem 1: $\psi_1(x) = 0$.

Algorithm	NI	NFE	$\|\eta\|$	$\|\psi_1(x_N)\|$	CPU Time [s]
PCNM4	6	24	2.957e-343	1.437e-1372	3.047e+00
PCNM8	6	30	1.003e-1186	5.724e-7120	2.484e+00
PJNM	7	21	3.185e-992	9.742e-3969	2.281e+00
KTNM	7	21	7.745e-454	3.124e-1814	2.156e+00
CNM1	10	30	8.112e-385	5.224e-770	3.093e+00
CNM2	10	30	8.112e-385	5.224e-770	3.219e+00
ONM	5	25	1.260e-732	1.890e-5859	2.719e+00

Problem 2. *NASA's "Wind" Satellite's Orbit. NASA intends to launch a spacecraft named Wind that will remain in a fixed position along a line from the earth to the sun so that the solar wind will travel by the satellite on its approach to the earth [30]. We devise the following equation based on the relevant physical laws:*

$$\psi_2(r) = r^3(R-r)^2\omega^2 - GM_s(R-r)^2 + GM_e r^2, \tag{5.3}$$

where, $G = 6.67 \times 10^{-11}$ *[Nm²/kg²],* $M_s = 1.98 \times 10^{30}$*[kg],* $M_e = 5.98 \times 10^{24}$*[kg], m = the mass of satellite [kg],* $R = 1.49 \times 10^{11}$*[m],* $T = 3.15576 \times 10^7$*[s] and* $\omega = \frac{2\pi}{T}$*. The required solution to the non-linear real-world problem* $\psi_2(r) = 0$ *is* $r^* = 1.4761775009615046593 \times 10^{11}$.

The numerical simulations of the real-life application problem carried out in Table 6 take a fixed number of iterations, meaning NI = 11 and $x_0 = 7.6$. It is observed that the existing fourth-order

PJNM algorithm produces, at the final iteration, the smallest absolute error and the smallest absolute value of the function itself when compared to the other algorithms. However, the proposed fourth-order algorithm PCNM4 has the advantage of less CPU time consumption in comparison to other fourth-order existing algorithms. Similarly, by fixing an initial guess of $x_0 = 5.5$, the numerical results are presented in Table 7. It is observed that the proposed PCNM8 and existing algorithm ONM (both having eighth-order convergence) take the fewest number of iterations in comparison to the other algorithms, whereas the existing eight-order ONM algorithm produces the smallest absolute error when compared to the other algorithms. We further observe that the PCNM4 and PJNM fourth-order algorithms takes the fewest number iteration (NI) in comparison to the other fourth-order algorithms KJNM, CNM1, and CNM2.

Table 6. Numerical simulations for Problem 2 for initial guess ($x_0 = 7.6$) and number of iterations (NI= 11).

Algorithm	$\|\eta\|$	$\|\psi_2(r_N)\|$	CPU Time [s]
PCNM4	1.3454e-810	1.6763e-3239	1.37e-01
PJNM	9.2337e-997	2.4946e-3984	1.88e-01
KTNM	1.7197e-155	2.0906e-618	1.56e-01
CNM1	4.1304e-35	6.6419e-51	1.40e-01
CNM2	4.1304e-35	6.6419e-51	1.40e-01

Table 7. Numerical simulations for Problem 2: $\psi_2(r) = 0$.

Algorithm	NI	NFE	$\|\eta\|$	$\|\psi_2(r_N)\|$	CPU Time [s]
PCNM4	11	44	1.345e-810	1.676e-324	2.578e+00
PCNM8	8	40	8.254e-727	2.333e-4375	8.430e-01
PJNM	11	33	9.234e-997	2.495e-3984	9.070e-01
KTNM	12	36	1.185e-646	4.701e-2583	9.540e-01
CNM1	14	42	2.156e-343	1.801e-667	7.820e-01
CNM2	14	42	2.156e-343	1.801e-667	7.810e-01
ONM	8	40	1.02e-1078	3.290e-8660	1.047e+00

Problem 3. *The system of nonlinear equations from [31] is:*

$$\begin{aligned} x_1 + \exp(x_2) - \cos(x_2) &= 0, \\ 3x_1 - x_2 - \sin(x_1) &= 0. \end{aligned} \tag{5.4}$$

The numerical simulations for solving the system numerical result carried out in Table 8 take a fixed number of iterations, that is $NI = 9$, and initial guess for two by two non linear system are $x_1 = 0.1$ and $x_2 = 0.2$. It is observed that the proposed algorithm PCNM4 produces the smallest absolute error when compared to the other algorithms. The proposed fourth-order algorithm PCNM4 has the advantage of less CPU time consumption in comparison to other fourth-order existing algorithms.

Table 8. Numerical simulations for Problem 3.

Algorithm	$\|\mathbf{x_n} - \mathbf{x_{n-1}}\|_\infty$	$[x_1, x_2]^T$	CPU Time [s]
PCNM4	2.00e-24002	6.77e-12003, 1.35e-12002	1.50e-02
PJNM	2.0354e-2313	-6.9050e-4627, -1.3810e-4626	3.12e-01
KTNM	1.7120e-4255	1.1777e-8510, 1.2507e-8510	2.97e-01
CNM1	8.6509e-1102	-5.2950e-2203 ,-1.0602e-2202	3.43e-01
CNM2	2.9729e-1084	-6.6197e-2168, -1.3258e-2167	3.44e-01

Problem 4. *The system of three nonlinear equations is [32]:*

$$\begin{aligned} 15x_1 + x_2^2 - 4x_3 - 13 = 0, \\ x_1^2 + 10x_2 - \exp(-x_3) - 11 = 0, \\ x_2^2 - 25x_3 + 22 = 0. \end{aligned} \tag{5.5}$$

The numerical simulations for solving the system (5.5) carried out in Table 9 take a fixed number of iterations, that is $NI = 6$, and the initial guesses for the three-by-three nonlinear system are chosen to be $x_1 = 0.8$, $x_2 = 1.0$, $x_3 = 0.8$. Although all the algorithms converge, we observe that the proposed algorithm PCNM4 produces the smallest absolute error when compared to the other algorithms. The proposed fourth-order algorithm PCNM4 has the advantage of less CPU time consumption in comparison to other fourth-order existing algorithms.

Table 9. Numerical simulations for Problem 4.

Algorithm	$\|\mathbf{x_n} - \mathbf{x_{n-1}}\|_\infty$	$[x_1, x_2, x_3]^T$	CPU Time[s]
PCNM4	1.91e-1754	1.04e+00 ,1.0300e+00,9.2300e-01	1.72e-01
PJNM	2.5908e-08	1.0418e+00 ,1.0312e+00 ,9.2254e-01	2.35e-01
KTNM	1.2176e-52	1.0418e+00, 1.0312e+00, 9.2254e-01	1.88e-01
CNM1	2.3203e-08	1.0418e+00, 1.0312e+00, 9.2254e-01	2.66e-01
CNM2	2.2706e-08	1.0418e+00, 1.0312e+00, 9.2254e-01	2.34e-01

Problem 5. *Neurophysiology application [33]:*

The nonlinear model consists of the following six equations:

$$\begin{aligned} x_1^2 + x_3^2 = 1, \\ x_2^2 + x_4^2 = 1, \\ x_5 x_3^3 + x_6 x_4^3 = c_1, \\ x_5 x_1^3 + x_6 x_2^3 = c_2, \\ x_5 x_1 x_3^2 + x_6 x_4^2 x_2 = c_3, \\ x_5 x_1^2 x_3 + x_6 x_2^2 x_4 = c_4. \end{aligned} \tag{5.6}$$

The constants c_i in (5.6) *can be randomly chosen. In our experiment, we considered $c_i = 0$ for $i = 1, \ldots, 4$.*

The numerical simulations for solving the real-life application system used in medical science (Neurophysiology) are carried out in Table 10. Neurophysiology is the branch of physiology that

focuses on the study of the nervous system, including the brain, spinal cord, and peripheral nerves. The numerical simulations carried out in Table 10 take a fixed number of iterations, that is $NI = 7$, and the initial guesses for the system given in (5.6) are chosen to be $x_1 = 0.9$, $x_2 = 0.8$, $x_3 = 0.7$, $x_4 = 0.5$, $x_5 = 0.3$, $x_6 = 0.1$. It is observed that the proposed algorithm, PCNM4, produces the smallest absolute error when compared to the other algorithms. The existing fourth-order algorithms taken for comparison, namely; PJNM, CNM1, and CNM2 diverge from the exact solution, while the algorithm KTNM converges but is comparatively slower than the proposed one. Hence, the performance of the proposed fourth-order algorithm for the nonlinear model (5.6) is far better than some of the existing approaches.

Table 10. Numerical simulations for Problem 5.

Algorithm	$\|\mathbf{x_n} - \mathbf{x_{n-1}}\|_\infty$	$[x_1, x_2, x_3, \ldots, x_6]^T$
PCNM4	1.54e-4839	7.89e-01, 8.48e-01, 6.14e-01, 5.30e-01, 3.06e-19384, 9.00e-21684
PJNM	7.4364e+05	7.88e-01 ,8.48e-01, 6.13e-01, 5.30e-01, -2.53e+05 ,-8.29e+05
KTNM	5.021e-30	7.99e-01, 8.63e-01, 6.01e-01 5.05e-01, -5.19e-59, 7.56e-60
CNM1	4.30e+12	5.75e+08, 4.99e+08, 5.66e+09 ,-4.51e+08 ,4.38e+12 ,-2.68e+12
CNM2	1.07e+4129	-8.4e+4121,3.1e+4123 ,1.0e+4122, -4.9e+4123,-1.4e+4128 ,1.0e+4129

6. Conclusions and future remarks

We introduce two predictor-corrector algorithms, amalgamating Taylor's series and the composition approach. One of these algorithms demonstrates eighth-order convergence alongside a high-efficiency index of approximately 1.5157, surpassing certain established techniques. The other algorithm achieves fourth-order convergence. Our comprehensive convergence analyses encompass both local and semilocal aspects. We use a number of complex polynomials and polynomiographs that show how much faster the modified algorithms converge, which is especially clear when we compare basins of attraction to other algorithms. Real-world applications in chemistry, astronomy, and neurology validate our simulations, where the proposed algorithms consistently outperform their numerical counterparts. It is intended to return to the proposed approach in the future to make any necessary adjustments to attain the best potential algorithm.

Author contributions

D. K. Almutairi: Validation, writing – review and editing; I. K. Argyros: Methodology, formal analysis, writing – review and editing; K. Gdawiec: Formal analysis, software, visualization, writing – original draft, writing – review and editing; S. Qureshi: Conceptualization, software, writing – original draft; A. Soomro: Software, investigation, writing – original draft; K. H. Jamali: Supervision, validation; M. Alquran: Supervision, resources, data curation; A. Tassaddiq: Conceptualization, methodology, validation, formal analysis, investigation, resources, writing – review and editing, project administration, funding acquisition. All authors have read and approved the final version of the manuscript for publication.

Acknowledgments

The author extends the appreciation to the Deanship of Postgraduate Studies and Scientific Research at Majmaah University for funding this research work through the project number (ER-2024-1216).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. T. J. Ypma, Historical development of the Newton-Raphson method, *SIAM Rev.*, **37** (1995), 531–551.
2. H. Ramos, J. Vigo-Aguiar, The application of Newton's method in vector form for solving nonlinear scalar equations where the classical newton method fails, *J. Comput. Appl. Math.*, **275** (2015), 228–237. https://doi.org/10.1016/j.cam.2014.07.028
3. H. Ramos, M. T. T. Monteiro, A new approach based on the Newton's method to solve systems of nonlinear equations, *J. Comput. Appl. Math.*, **318** (2017), 3–13. https://doi.org/10.1016/j.cam.2016.12.019
4. H. A. Abro, M. M. Shaikh, A new time-efficient and convergent nonlinear solver, *Appl. Math. Comput.*, **355** (2019), 516–536. https://doi.org/10.1016/j.amc.2019.03.012
5. S. Qureshi, I. K. Argyros, A. Soomro, K. Gdawiec, A. A. Shaikh, E. Hincal, A new optimal root-finding iterative algorithm: Local and semilocal analysis with polynomiography, *Numer. Algorithms*, **95** (2024), 1715–1745. https://doi.org/10.1007/s11075-023-01625-7
6. A. Naseem, M. A. Rehman, S. Qureshi, N. A. D. Ide, Graphical and numerical study of a newly developed root-finding algorithm and its engineering applications, *IEEE Access*, **11** (2023), 2375–2383. https://doi.org/10.1109/ACCESS.2023.3234111
7. R. Behl, A. Cordero, S. S. Motsa, J. R. Torregrosa, An eighth-order family of optimal multiple root finders and its dynamics, *Numer. Algorithms*, **77** (2018), 1249–1272. https://doi.org/10.1007/s11075-017-0361-6
8. A. Soomro, A. Naseem, S. Qureshi, N. A. D. Ide, Development of a new multi-step iteration scheme for solving non-linear models with complex polynomiography, *Complexity*, **2022** (2022). https://doi.org/10.1155/2022/2596924
9. H. Ahmad, D. U. Ozsahin, U. Farooq, M. A. Fahmy, M. D. Albalwi, H. Abu-Zinadah, Comparative analysis of new approximate analytical method and Mohand variational transform method for the solution of wave-like equations with variable coefficients, *Results Phys.*, **51** (2023), 106623. https://doi.org/10.1016/j.rinp.2023.106623
10. K. K. Ali, S. Tarla, A. Yusuf, Quantum-mechanical properties of long-lived optical pulses in the fourth-order KdV-type hierarchy nonlinear model, *Opt. Quant. Electron.*, **55** (2023), 590. https://doi.org/10.1007/s11082-023-04817-6

11. M. Ozisik, A. Secer, M. Bayram, A. Yusuf, T. A. Sulaiman, Soliton solutions of the (2 + 1)-dimensional Kadomtsev-Petviashvili equation via two different integration schemes, *Int. J. Mod. Phys. B*, **37** (2023), 2350212. https://doi.org/10.1142/S0217979223502120
12. T. A. Sulaiman, A. Yusuf, A. S. Alshomrani, D. Baleanu, Wave solutions to the more general (2 + 1)-dimensional Boussinesq equation arising in ocean engineering, *Int. J. Mod. Phys. B*, **37** (2023), 2350214. https://doi.org/10.1142/S0217979223502144
13. J. L. Hueso, E. Martínez, C. Teruel, Multipoint efficient iterative methods and the dynamics of Ostrowski's method, *Int. J. Comput. Math.*, **96** (2019), 1687–1701. https://doi.org/10.1080/00207160.2015.1080354
14. R. Behl, A. Cordero, S. S. Motsa, J. R. Torregrosa, Construction of fourth-order optimal families of iterative methods and their dynamics, *Appl. Math. Comput.*, **271** (2015), 89–101. https://doi.org/10.1016/j.amc.2015.08.113
15. A. Y. Özban, B. Kaya, A new family of optimal fourth-order iterative methods for nonlinear equations, *Results Control Optim.*, **8** (2022), 100157. https://doi.org/10.1016/j.rico.2022.100157
16. B. Kong-ied, Two new eighth and twelfth order iterative methods for solving nonlinear equations, *Int. J. Math. Comput. Sci.*, **16** (2021), 333–344.
17. D. Herceg, D. Herceg, Eighth order family of iterative methods for nonlinear equations and their basins of attraction, *J. Comput. Appl. Math.*, **343** (2018), 458–480. https://doi.org/10.1016/j.cam.2018.04.040
18. J. Kou, Y. Li, X. Wang, Fourth-order iterative methods free from second derivative, *Appl. Math. Comput.*, **184** (2007), 880–885. https://doi.org/10.1016/j.amc.2006.05.189
19. I. K. Argyros, *Approximate solution of operator equations with applications*, Singapore: World Scientific, 2005. https://doi.org/10.1142/5851
20. S. Regmi, I. K. Argyros, S. George, C. I. Argyros, Extended semilocal convergence for Chebyshev-Halley-type schemes for solving nonlinear equations under weak conditions, *Contemp. Math.*, **4** (2023), 1–16.
21. I. K. Argyros, R. Behl, S. S. Motsa, Unifying semilocal and local convergence of Newton's method on Banach space with a convergence structure, *Appl. Numer. Math.*, **115** (2017), 225–234. https://doi.org/10.1016/j.apnum.2017.01.008
22. I. K. Argyros, S. George, Ball convergence of Newton's method for generalized equations using restricted convergence domains and majorant conditions, *Nonlinear Funct. Anal. Appl.*, **22** (2017), 485–494.
23. I. K. Argyros, Results on Newton methods: Part II. perturbed Newton-like methods in generalized Banach spaces, *Appl. Math. Comput.*, **74** (1996), 143–159. https://doi.org/10.1016/0096-3003(95)00118-2
24. I. Gościniak, K. Gdawiec, Control of dynamics of the modified Newton-Raphson algorithm, *Commun. Nonlinear Sci. Numer. Simul.*, **67** (2019), 76–99. https://doi.org/10.1016/j.cnsns.2018.07.010
25. I. Petković, L. Z. Rančić, Computational geometry as a tool for studying root-finding methods, *Filomat*, **33** (2019), 1019–1027. https://doi.org/10.2298/FIL1904019P

26. B. Kalantari, *Polynomial root-finding and polynomiography*, Singapore: World Scientific, 2009. https://doi.org/10.1142/6265

27. G. Ardelean, O. Cosma, L. Balog, A comparison of some fixed point iteration procedures by using the basins of attraction, *Carpathian J. Math.*, **32** (2019), 277–284.

28. K. Gdawiec, W. Kotarski, A. Lisowska, On the robust Newton's method with the Mann iteration and the artistic patterns from its dynamics, *Nonlinear Dyn.*, **104** (2021), 297–331. https://doi.org/10.1007/s11071-021-06306-5

29. H. Sharma, M. Kansal, R. Behl, An efficient two-step iterative family adaptive with memory for solving nonlinear equations and their applications, *Math. Comput. Appl.*, **27** (2022), 97. https://doi.org/10.3390/mca27060097

30. W. Y. Yang, W. Cao, J. Kim, K. W. Park, H. H. Park, J. Joung, et al., *Applied numerical methods using MATLAB*, 2 Eds., Hoboken: John Wiley & Sons, 2020.

31. J. R. Sharma, R. Sharma, N. Kalra, A novel family of composite Newton-Traub methods for solving systems of nonlinear equations, *Appl. Math. Comput.*, **269** (2015), 520–535. https://doi.org/10.1016/j.amc.2015.07.092

32. R. L. Burden, *Numerical analysis*, Brooks/Cole Cengage Learning, 2011.

33. C. Grosan, A. Abraham, A new approach for solving nonlinear equations systems, *IEEE Trans. Syst. Man, Cy. A*, **38** (2008), 698–714. https://doi.org/10.1109/TSMCA.2008.918599

AIMS Press