

Article

Improving the Convergence of Interval Single-Step Method for Simultaneous Approximation of Polynomial Zeros

Nur Raidah Salim ^{1,*},[†] , Chuei Yee Chen ^{1,2,†} , Zahari Mahad ¹  and Siti Hasana Sapar ^{1,2}

¹ Institute for Mathematical Research, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia; cychen@upm.edu.my (C.Y.C.); zaharimahad@upm.edu.my (Z.M.); sitihas@upm.edu.my (S.H.S.)

² Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia

* Correspondence: nurraidah@upm.edu.my

† These authors contributed equally to this work.

Abstract: This paper describes the extended method of solving real polynomial zeros problems using the single-step method, namely, the interval trio midpoint symmetric single-step (ITMSS) method, which updates the midpoint at each forward-backward-forward step. The proposed algorithm will constantly update the value of the midpoint of each interval of the previous roots before entering the preceding steps; hence, it always generate intervals that decrease toward the polynomial zeros. Theoretically, the proposed method possesses a superior rate of convergence at 16, while the existing methods are known to have, at most, 9. To validate its efficiency, we perform numerical experiments on 52 polynomials, and the results are presented, using performance profiles. The numerical results indicate that the proposed method surpasses the other three methods by fine-tuning the midpoint, which reduces the final interval width upon convergence with fewer iterations.

Keywords: single-step method; interval arithmetic; R-order of convergence; performance profile; polynomial zeros



Citation: Salim, N.R.; Chen, C.Y.; Mahad, Z.; Sapar, S.H. Improving the Convergence of Interval Single-Step Method for Simultaneous Approximation of Polynomial Zeros. *Symmetry* **2021**, *13*, 1971. <https://doi.org/10.3390/sym13101971>

Academic Editor: Clemente Cesarano

Received: 20 September 2021

Accepted: 14 October 2021

Published: 19 October 2021

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



Copyright: © 2021 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

The widespread application of interval arithmetic was inhibited in the past by a lack of hardware and software. Nonetheless, more real-world applications have appeared in recent years. In the twenty-first century, interval arithmetic was discovered to have substantially contributed to the steganography field, notably in improving the quality of watermarked images [1]. Interval arithmetic also plays a significant role in enhancing the power of higher performance computing, such as GPUs [2]. Furthermore, in the era of data science and artificial intelligence, many scholars and practitioners from various backgrounds have incorporated the interval analysis concepts into their existing models or methods in order to investigate uncertainty propagation in specific data or systems [3–5]. These applications have led to increased attention and more rigorous studies on interval methods over the recent years.

The history of simultaneous inclusion of polynomial zeros can be traced back to the Weierstrass' function [6], where the iterative procedures for finding polynomial zeros is guaranteed to be of quadratic convergence. Numerous studies by many scholars have contributed to the development of polynomial inclusion studies over the decades. In recent years, the approach for bounding polynomial zeroes simultaneously was studied via different strategies, such as the Chebyshev-like root-finding method [7], Weierstrass root-finding method [8,9], Halley's method [10], Ehrlich method [11,12] and also by considering various types of initial conditions to solve this problem [13]. In 2014, Proinov and Petcova [14] obtained the important result related to the semi-local convergence of the Weierstrass root finding method. Later, Cholakov and Vasileva [15] introduced and

studied a new fourth-order iterative method for finding all zeros of a polynomial simultaneously to achieve semi-local convergence. Correspondingly, Kyncheva et al. [16] also transformed the convergence theorem of Newton, Halley, and Chebychev into semi-local convergence theorems for simultaneous determination of multiple polynomial zeroes with accurate initial conditions. In other respects, Proinov and Ivanov [17] elaborated in detail the analysis of any local convergence of the Sakurai–Torii–Sugiura method specifically for high-order iterative problems in finding that polynomial zeroes can be transformed into a semi-local convergence.

Among the classical schemes for finding polynomial roots in the sense of interval arithmetic are those by Gargantini, and Henrici [18], and Petkovic [19]. In 1988, Monsi and Wolfe [20] stated that instead of using the standard point single-step method [21], which involved specific points in the algorithm, they proposed to tackle the problem by computing sets on the real line that exactly resembled the fundamental of intervals arithmetic [22]. The point iterative methods might be very efficient but somehow have many disadvantages, such as that the sequence obtained only converges for perfect initial estimates of the zeros. On the other hand, several modifications on the point total-step method using interval has improved the order of convergence and computational analysis in terms of the number of iterations, such as the interval single-step (IS1) procedure [23], and interval single-step (IS2) procedure [24]. The difference between these two methods is discussed in Section 2. Later, Salim et al. collaborated the idea of [20,21] and [24] by proving that the new extension approach known as the interval single-step (ISS2) method [25], has a superior order of convergence ($R = 9$) and lesser iterations generated. In 2018, Jamaludin et al. [26] showed a significant reduction in the computational time when compared to its prior modifications, somehow; the R -order of convergence was not clear and has not been discussed by the authors.

Therefore, this paper proposes an interval iterative procedure for finding polynomial zeros with superior convergence. This method imposes that the value of midpoints computed for use in the first loop are renewed in the next loop. While the standard interval single-step procedures compute the midpoint only once at each iteration, our procedure updates the midpoints at each step, leading to reduced iterations with smaller final intervals.

This paper is organized accordingly as follows. In Section 2, we first present the standard formulation of finding the root problems that lead to interval single-step procedures. This is then followed by establishing our proposed iterative procedure, and we end the section by proving its inclusion property. Next, in Section 3, the convergence analysis and numerical results are displayed, using the performance profile to compare the efficiency of the methods. The following section includes sufficient arguments based on the findings. Finally, Section 5 provides a concise conclusion.

2. Materials and Method

In this section, we will discuss the standard formulation of finding the root problem.

2.1. Interval Single-Step Method

We begin this section with the class of simultaneous methods, which leads to our proposed method. Let $p : \mathbb{R} \rightarrow \mathbb{R}$ be a polynomial of degree n :

$$p(x) = \sum_{i=0}^n a_i x^i \quad (1)$$

where $a_i \in \mathbb{R}$, $i = 0, \dots, n$ are given and $a_n \neq 0$. A zero of polynomials is equivalent to a root of the equation $p(x) = 0$.

If polynomial (1) has distinct zeros $x_i^*, i = 1, 2, \dots, n$, then it can be written as follows:

$$p(x) = \prod_{j=1}^n (x - x_j^*) \quad (2)$$

by letting $a_n = 1$. It follows from (2) that any zero can be expressed as follows:

$$x_i^* = x_i - \frac{p(x_i)}{\prod_{j \neq i} (x_i - x_j^*)}. \quad (3)$$

If $x_j \approx x_j^* (j = 1, \dots, n)$ so by (3), we have

$$x_i^* \approx x_i - \frac{p(x_i)}{\prod_{j \neq i} (x_i - x_j)}, \quad i = 1, \dots, n.$$

This gives us the point total-step iterative procedure defined by the following:

$$x_i^{(k+1)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j=1, j \neq i}^n (x_i^{(k)} - x_j^{(k)})}, \quad i = 1, \dots, n; k = 0, 1, 2, \dots \quad (4)$$

where $x_i^{(0)}$ is some initial guess. The value $x_i^{(k+1)}$ can be updated one at a time or simultaneously. It is firstly mentioned by Weierstrass [6], Durand [27] and Kerner [28], which is also known as the WDK method. When compared to Newton's method, the WDK method is much more robust, i.e., regardless of the initial assumption, it converges to the zeros approximately.

The simultaneous computation of the polynomial zeros in the sense of (4) was proposed by [21] and is known as the point single-step (PS1):

$$x_i^{(k+1)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j=1}^{i-1} (x_i^{(k)} - x_j^{(k+1)}) \prod_{j=i+1}^n (x_i^{(k)} - x_j^{(k)})}, \quad i = 1, \dots, n; k = 0, 1, 2, \dots \quad (5)$$

Due to its simplicity in implementation, the PS1 has been studied and extended in various ways. For example, Alefeld and Herzberger [23] improvised PS1 using the interval approach and named it the interval single-step (IS1) method. Later, the modifications of the variants were tested for five polynomials, producing a fewer number of iterations and quicker CPU time [29]. In addition, Chen et al. [30] demonstrated that adding a scaling function to IS1 outperforms the existing procedures, leading to a more significant reduction in the final interval width with fewer iterations.

An alternative expression of (5) is as follows. We first differentiate (2) with respect to x to give the following:

$$p'(x) = \sum_{i=1}^n \prod_{j \neq i}^n (x - x_j^*). \quad (6)$$

From (2) and (6), we have the following:

$$\frac{p'(x_i)}{p(x_i)} = \sum_{j=1}^n \frac{1}{x_i - x_j^*} = \frac{1}{x_i - x_i^*} + \sum_{j \neq i}^n \frac{1}{x_i - x_j^*}. \quad (7)$$

Rearranging Equation (7), we obtain the following:

$$\frac{1}{x_i - x_i^*} = \frac{p'(x_i)}{p(x_i)} - \sum_{j \neq i}^n \frac{1}{x_i - x_j^*}$$

and therefore

$$x_i - x_i^* = \frac{1}{\frac{p'(x_i)}{p(x_i)} - \sum_{j \neq i}^n \frac{1}{x_i - x_j^*}}. \quad (8)$$

Finally, we have the following equation:

$$x_i^* = x_i - \frac{1}{\frac{p'(x_i)}{p(x_i)} - \sum_{j \neq i}^n \frac{1}{x_i - x_j^*}}. \quad (9)$$

By implementing (4) to (6) and (9), a revised point single-step (PS2) method [21] can be expressed as follows:

$$x_i^{(k+1)} = x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)} \left(\sum_{j=1}^{i-1} \frac{1}{x_i^{(k)} - x_j^{(k+1)}} + \sum_{j=i+1}^n \frac{1}{x_i^{(k)} - x_j^{(k)}} \right) \right]}, \quad i = 1, \dots, n; k = 0, 1, 2, \dots \quad (10)$$

where

$$g_i^{(k)} = g(x_i^{(k)}) = \frac{p(x_i^{(k)})}{p'(x_i^{(k)})}.$$

Incorporating the interval computation into (10), Salim [24] proposed the following iterative formula:

$$X_i^{(k+1)} = x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)} \left(\sum_{j=1}^{i-1} \frac{1}{x_i^{(k)} - X_j^{(k+1)}} + \sum_{j=i+1}^n \frac{1}{x_i^{(k)} - X_j^{(k)}} \right) \right]}, \quad i = 1, \dots, n; k = 0, 1, 2, \dots \quad (11)$$

where $x_i^{(k)}$ is the midpoint of the interval $X_i^{(k)}$. Known as the alternative interval single step procedure (IS2), the R-order of convergence of the iterative procedure (11) is more than 3. The proof of this holds with the corresponding proof of the PS2 method and is almost identical. Later, Salim et al. [25] improved the corresponding R-order of convergence of the ISS2 method, which is at least 9.

2.2. Interval Trio Symmetric Single-Step (ITMSS) Method

This study proposes the interval trio midpoint symmetric single-step (ITMSS) method for bounding real zeros of a polynomial simultaneously. The proposed method will update each interval's midpoint value of X , denoted by $mid(X)$, and revise the value of $g_i^{(k)}$ before entering the next step. Enforcing this strategy will allow us to narrow the computed bounds rigorously. The process is repeated until smaller intervals with guaranteed roots are generated, and the stopping condition imposed on the interval width is satisfied. We denote the width of interval X as $w(X)$. Table 1 describes the proposed algorithm in detail.

Table 1. Interval trio symmetric single-step (ITMSS).

Step 0:	Given initial intervals $X_1^{(0)}, X_2^{(0)}, \dots, X_n^{(0)}$ and $X_i^{(0)} \cap X_j^{(0)} = \emptyset, i \neq j$. Set the stopping criterion ε
Step 1:	For $k \geq 0$, set $x_i^{(k)} = \text{mid}(X_i^{(k)}), i = 1, \dots, n$. Compute $g_i^{(k)} = g(x_i^{(k)}) = \frac{p(x_i^{(k)})}{p'(x_i^{(k)})}$.
Step 2.1:	Compute $X_i^{(k,1)} = \left\{ x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)} \left(\sum_{j=1}^{i-1} \frac{1}{x_i^{(k)} - X_j^{(k,1)}} + \sum_{j=i+1}^n \frac{1}{x_i^{(k)} - X_j^{(k)}} \right) \right]} \right\} \cap X_i^{(k)}, i = 1, \dots, n,$ $x_i^{(k,1)} = \text{mid}(X_i^{(k,1)}), g^{(k,1)} = \frac{p(x_i^{(k,1)})}{p'(x_i^{(k,1)})}$
Step 2.2:	Compute $X_i^{(k,2)} = \left\{ x_i^{(k,1)} - \frac{g_i^{(k,1)}}{\left[1 - g_i^{(k,1)} \left(\sum_{j=1}^{i-1} \frac{1}{x_i^{(k,1)} - X_j^{(k,1)}} + \sum_{j=i+1}^n \frac{1}{x_i^{(k,1)} - X_j^{(k,2)}} \right) \right]} \right\} \cap X_i^{(k,1)}, i = n, \dots, 1,$ $x_i^{(k,2)} = \text{mid}(X_i^{(k,2)}), g^{(k,2)} = \frac{p(x_i^{(k,2)})}{p'(x_i^{(k,2)})}$
Step 2.3:	Compute $X_i^{(k,3)} = \left\{ x_i^{(k,2)} - \frac{g_i^{(k,2)}}{\left[1 - g_i^{(k,2)} \left(\sum_{j=1}^{i-1} \frac{1}{x_i^{(k,2)} - X_j^{(k,3)}} + \sum_{j=i+1}^n \frac{1}{x_i^{(k,2)} - X_j^{(k,2)}} \right) \right]} \right\} \cap X_i^{(k,2)}, i = 1, \dots, n,$ $x_i^{(k,3)} = \text{mid}(X_i^{(k,3)}), g^{(k,3)} = \frac{p(x_i^{(k,3)})}{p'(x_i^{(k,3)})}$
Step 2.4:	Set $X_i^{(k+1)} = X_i^{(k,3)}$
Step 3:	If $w(X_i^{(k+1)}) < \varepsilon$, for every $i = 1, 2, \dots, n$ then stop. Else, set $k = k + 1$ and $g^{(k+1)} = g^{(k,3)}$, and go to Step 1.

The significance of the ITMSS algorithm is that the values of $g_i^{(k)}, i = 1, \dots, n$ in Step 1, which are computed for use in Step 2.1, are renewed every time it completes the inner loop of Step 2. This means that we will compute the new midpoint values from the final interval width for the used in the next internal loop of Step 2, generating the updating values of $g_i^{(k,1)}, g_i^{(k,2)}$, and $g_i^{(k,3)}$ for every root, i , where $i = 1, \dots, n$. It also has the following forward-backward-forward attractive features, where the value of summations $\sum_{j=1}^{i-1} \frac{1}{x_i^{(k)} - X_j^{(k,1)}}, i = 2, \dots, n$, which are computed in Step 2.1, will be used in Step 2.2. Hence, it will constantly update the value of the midpoint of each interval of the previous roots before entering the following steps that will always generate intervals that decrease toward the polynomial zeros.

Moreover, the value from summations $\sum_{j=1}^{i-1} \frac{1}{x_i^{(k)} - X_j^{(k,2)}}, i = 2, \dots, n$, which are computed in Step 2.2, will be used in Step 2.3. The iteration stops when $w(X_i^{(k+1)}) < \varepsilon$ for some fixed stopping criterion $\varepsilon = 10^{-16}$. Otherwise, set $k = k + 1$ and then subsequently $X_i^{(k+1)} = X_i^{(k,3)}$ and $g^{(k+1)} = g^{(k,3)}$. The iteration stops when the stopping condition is satisfied. The renewing of midpoint values before computing the new $g_i^{(k)}$ are repeated three times in the internal loop to increase the convergence to the zeros simultaneously in each iteration, hence the name, interval trio midpoint symmetric single-step (ITMSS) method.

The following shows that the proposed method with updating the midpoint of the enclosing intervals will always generate intervals that decrease toward the polynomial zeros.

Theorem 1. Let $p(x) = a^{(n)}x^n + a^{(n-1)}x^{n-1} + \dots + a^{(0)}$ be a polynomial with n simple roots $x_i^*, (1 \leq i \leq n)$. Inclusion intervals $X^{(0,j)} \ni \xi_i^*, 1 \leq i \leq n$, are furthermore known for which

$X^{(0,j)} \cap X^{(0,k)} = \emptyset$, $1 \leq j \leq k \leq n$ holds. It follows that the sequence $\{X^{(k)}\}_{k=0}^{\infty}$, $1 \leq i \leq n$, generated from the ITMSS Algorithm, satisfies the following:

$$x_i^* \in X^{(k,i)}, k \geq 0$$

and

$$X^{(0,i)} \supset X^{(1,i)} \supset X^{(2,i)} \supset \dots \text{ with } \lim_{k \rightarrow \infty} X^{(k,i)} = x_i^*$$

or the sequence comes to rest at $[x_i^*, x_i^*]$ after a finite number of steps.

Proof of Theorem 1. Let $X = [\underline{x}, \bar{x}]$. By substituting

$$m(X_i^{(k)}) = \frac{1}{2}(\underline{x}_i^{(k)} + \bar{x}_i^{(k)}) = x_i^*$$

in this ITMSS method and considering the construction of (11), it follows immediately that the width of the inclusions for each zero is at least halved at each calculation of a new iteration. $\square$

Theorem 1 partially holds when the polynomial has multiple roots. If we collect these multiple roots together as $x_1^*, x_2^*, \dots, x_n^*$, this approach must be altered so that the new calculations of the included intervals are only done for the indices $1 \leq i \leq n$. Theorem 1 is then only valid for simple zeros, where the included intervals are recomputed at each step. Meanwhile, the other intervals remain unchanged [21].

3. Results

In this section, we present the theoretical convergence results of the proposed method, which is then followed by numerical results on real polynomials.

3.1. Convergence Analysis

The following theorem is about the inclusion of the generated intervals $\{X_i^{(k)}\}$, $i = 1, \dots, n$ and their convergence to x_i^* , $i = 1, \dots, n$. Finally, we establish the R-order of convergence of the proposed method, and the theoretical analysis of convergence will be discussed.

Theorem 2. Let $I(\mathbb{R})$ be the set of all closed intervals on the real line and D_i be subset of $I(\mathbb{R})$ for $i = 1, \dots, n$. If the assumptions of Theorem 1 are valid, then it follows that $0 \notin D_i \in I(\mathbb{R})$ such that $p'(x) \in D_i$, $i = 1, \dots, n$, and then

$$w(X_i^{(k+1)}) \leq \frac{1}{2} \left(1 - \frac{d_{iL}}{d_{iS}} \right) w(X_i^{(k)})$$

are satisfied. Finally, the R-Order of convergence of ITMSS is given by the following:

$$\mathcal{O}_{\mathcal{R}}(\text{ITMSS}, x^*) \geq 16, \quad i = 1, \dots, n.$$

Proof of Theorem 2. The proof that $w(X_i^{(k+1)}) \leq \frac{1}{2} \left(1 - \frac{d_{iL}}{d_{iS}} \right) w(X_i^{(k)})$ and that $X_i^{(k)} \rightarrow x_i^*$, $k \rightarrow \infty$, $i = 1, \dots, n$ holds is almost identical with the corresponding proofs in [21], and is therefore omitted. It remains to be proven that R-order of convergence is at least 16.

As in the proof of Theorem [21], it may be shown that there exists $\alpha > 0$ such that for every $k \geq 0$, the following holds:

$$w_i^{(k,1)} \leq \beta \left(w_i^{(k,0)} \right)^5 \left\{ \sum_{j=1}^{i-1} w_j^{(k,1)} + \sum_{j=i+1}^n w_j^{(k,0)} \right\}, \quad i = 1, \dots, n \quad (12)$$

and

$$w_i^{(k,2)} \leq \beta \left(w_i^{(k,0)} \right)^5 \left\{ \sum_{j=i+1}^{i-1} w_j^{(k,1)} + \sum_{j=i+1}^n w_j^{(k,2)} \right\}, \quad i = n, \dots, 1 \quad (13)$$

and

$$w_i^{(k,3)} \leq \beta \left(w_i^{(k,0)} \right)^5 \left\{ \sum_{j=i+1}^{i-1} w_j^{(k,2)} + \sum_{j=i+1}^n w_j^{(k,3)} \right\}, \quad i = 1, \dots, n \quad (14)$$

where

$$w_i^{(k,s)} = 1(n-1)\alpha w \left(X_i^{(k,s)} \right), \quad s = 0, 1, 2, 3, \quad (15)$$

and

$$\beta = \frac{1}{n-1}.$$

Let the following hold:

$$u_i^{(1,1)} = \begin{cases} 6, & i = 1, \dots, n-1 \\ 11, & i = n \end{cases} \quad (16)$$

$$u_i^{(1,2)} = \begin{cases} 16, & i = 1 \\ 11, & i = 2, \dots, n-1 \end{cases} \quad (17)$$

$$u_i^{(1,3)} = \begin{cases} 16, & i = 1, \dots, n-1 \\ 21, & i = n \end{cases} \quad (18)$$

and for $r = 1, 2, 3$, with

$$u_i^{(k+1,r)} = \begin{cases} 16u_i^{(k)}, & i = 1, \dots, n-1 \\ 16u_i^{(k)} + 5^{(k+1)}, & i = n \end{cases}. \quad (19)$$

Then, by (16)–(19) for every $k \geq 0$, we have the following:

$$u_i^{(k,1)} = \begin{cases} 6(16^{(k-1)}), & i = 1, \dots, n-1 \\ 11(16^{(k-1)}) + \sum_{j=0}^{k-2} 5^{(k-j)} 16^j, & i = n \end{cases} \quad (20)$$

and

$$u_i^{(k,2)} = \begin{cases} 11(16^{(k-1)}) + \sum_{j=0}^{k-2} 5^{(k-j)} 16^j, & i = n \\ 11(16^{(k-1)}), & i = n-1, \dots, 2 \\ 16^{(k)}, & i = 1 \end{cases} \quad (21)$$

and

$$u_i^{(k,3)} = \begin{cases} 16^{(k)}, & i = 1, \dots, n-1 \\ 21(16^{(k-1)}) + \sum_{j=0}^{k-2} 5^{(k-j)} 16^j, & i = n \end{cases}. \quad (22)$$

Suppose without loss of generality, we have the following:

$$w_i^{(0,0)} \leq h < 1, \quad i = 1, \dots, n. \quad (23)$$

Then, by an inductive argument, it follows from (12)–(23) that for $i = 1, \dots, n$ and $k \geq 0$, we have the following:

$$w_i^{(k,1)} \leq h^{u_i^{(k+1,1)}},$$

and

$$w_i^{(k,2)} \leq h^{u_i^{(k+1,2)}},$$

and

$$w_i^{(k,3)} \leq h^{u_i^{(k+1,3)}}.$$

Hence, by (18) and Step 2.4, for every $k \geq 0$, we have the following:

$$w_i^{(k+1)} \leq h^{4^{2(k+1)}} = h^{4^{2k+2}}.$$

Thus, the following is true:

$$w^{(k)} \leq h^{4^{2k}} = h^{16^k}.$$

Then for every $k \geq 0$, by (12)–(23),

$$w(X_i^{(k)}) \leq \left(\frac{\beta}{\alpha}\right) h^{16^k}, \quad i = 1, \dots, n \quad (24)$$

Let the following hold:

$$w^{(k)} = \max_{1 \leq i \leq n} \{w(X_i^{(k)})\},$$

and by (24), we have the following:

$$w^{(k)} \leq \left(\frac{\beta}{\alpha}\right) h^{16^k}.$$

Hence,

$$\begin{aligned} R_{16}(w^{(k)}) &= \limsup_{k \rightarrow \infty} \left\{ \left(w^{(k)} \right)^{\frac{1}{(16^k)}} \right\} \\ &= \limsup_{k \rightarrow \infty} \left\{ \left(\frac{\beta}{\alpha} \right)^{\frac{1}{(16^k)}} h \right\} \\ &= h < 1. \end{aligned}$$

Therefore, it follows from [21] and [31] that, $\mathcal{O}_{\mathcal{R}}(ITMSS, x_i^*) \geq 16$, $i = 1, \dots, n$. $\square$

3.2. Numerical Experiments

We analyze the efficiency of the proposed method by comparing the computational results of the 52 test problems with the interval single-steps (IS2) method and its variants regarding the number of iterations and the largest final width of the interval generated [26,30,32]. The selected test examples are arranged starting from a real polynomial $p(x)$ with degree $n = 3$ up to $n = 12$. Furthermore, we only consider real and simple zeros in this experiment. Next, these algorithms are implemented using MATLAB R2017b cooperated with the Intlab V12.1 toolbox, specifically for interval arithmetic developed by Rump [33]. Meanwhile, the stopping criterion used is $w(X_i^{(k)}) \leq 10^{-16}$, $i = 1, \dots, n$. Then, we observe the results using the performance profile to test the degree of efficiency of our algorithms. The algorithms that are considered include the following:

1. Interval single-step (IS2) method [20];
2. Interval symmetric single-step (ISS2) method [25];
3. Interval zero-symmetric single-step (IZSS2) method [32];
4. Interval trio midpoint symmetric single-step (ITMSS) method.

We begin with setting up the initial intervals for each n in each test example into all four algorithms stated above in order to compute the number of iterations, k , and the largest final interval width, $w(X_i^{(k)})$. There are 52 test examples that consist of 343 starting points, multiplied by four algorithms and then multiplied by two output categories (k and w). Thus, 416 of the output findings must be assessed using a performance profile. According to Dolan and Moré [34], when a large number of tested examples are employed, such as 100, 250, 500, or 1000, the output analyzing phases become extremely tough to evaluate, motivating researchers to apply performance profile comparison. In a nutshell, a performance profile is a visualization-based analytical tool used to evaluate the results of a benchmark experiment, allowing the user to compare the performance of each solver. Therefore, it is also known as a (cumulative) distribution function of a performance indicator, where $\rho(\tau)$ is the probability that a performance ratio is at most τ (the best possible ratio). The solver's likelihood will prevail over the other solvers, and is represented by $\rho(1)$. If we are mainly concerned about the number of triumphs, we can compare the values of $\rho(1)$ for each solver. Hence, the preferable solver is the one with the highest number of winning results.

The performance profiles for the number of iterations, k , and the maximum final interval width, w , are shown in Figures 1 and 2, respectively. The lines indicate the methods on the graphs. From Figure 1, it is noticeable that the ITMSS method performs better than IS2, ISS2, and IZSS2, respectively. In other words, the midpoint renewing procedures require fewer iterations, and converge to the zeros in terms of the number of iterations. As previously stated, the ITMSS method has a better chance of winning and is the most preferred method of all. Note that the order of convergence for IS2 is at least 3 [24], and the graph shows that this method is the least efficient in terms of the number of iterations. Meanwhile, the order of convergence for ISS2 and IZSS2 is at least 9 and 13, respectively [25,32]. Therefore, from an overall view of Figure 1, the method's efficiency resembles the order of convergence of the methods.

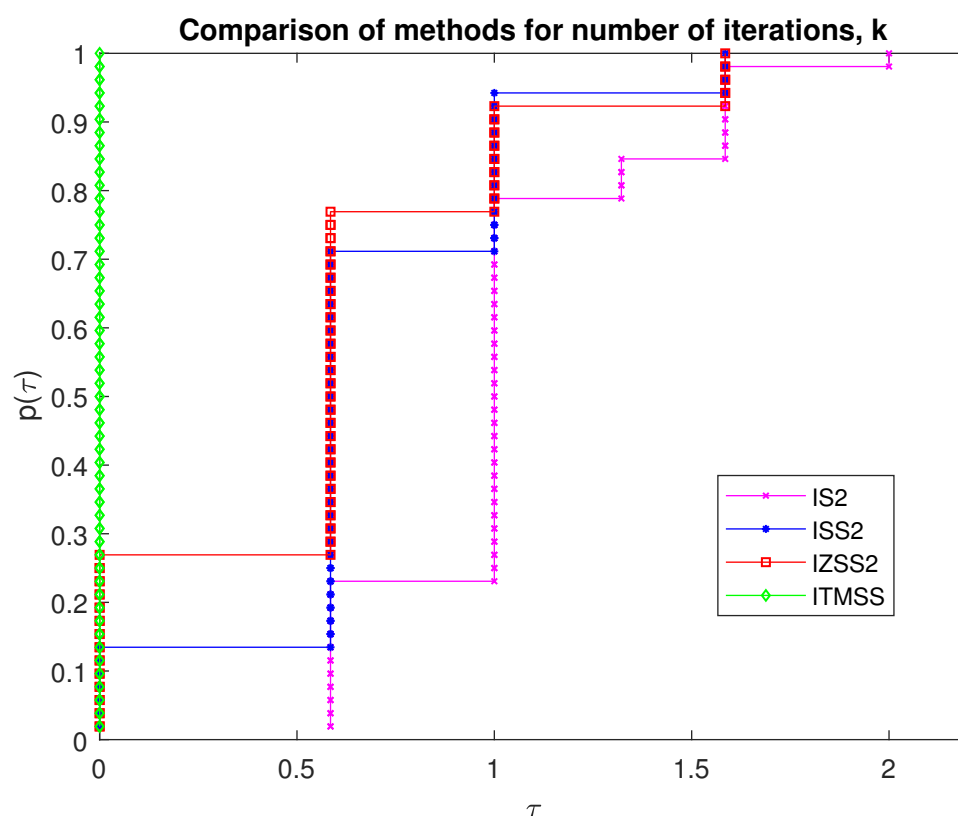


Figure 1. Comparison of methods for number of iterations, k .

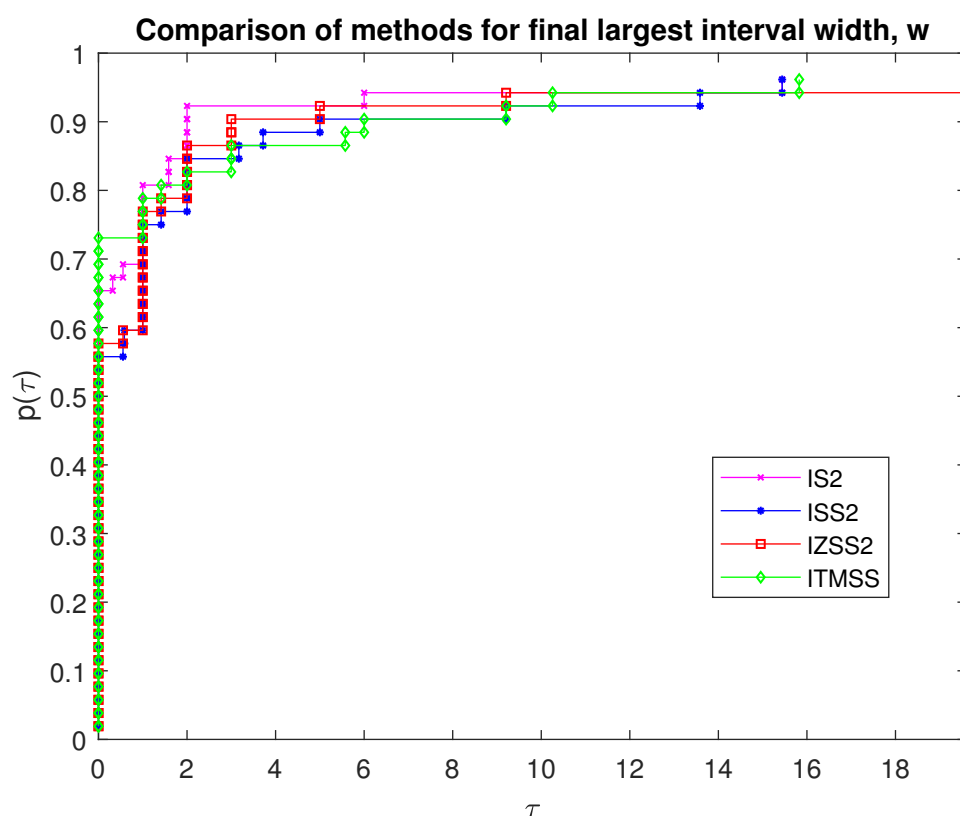


Figure 2. Comparison of methods for final largest interval width, w .

4. Discussion

In this study, we considered the polynomial zeros inclusion problem specifically by using a single-step procedure. The idea is to propose a more dynamic approach to finding the zeros. Considering that adjusting the midpoint of the intervals at every inner loop of the algorithm will encapsulate all real potential results, the interval arithmetic method's precision is guaranteed. It fulfills the stopping condition faster without neglecting other preliminary concepts of interval arithmetic. Based on the idea, we name the proposed method as the interval trio midpoint symmetric single-step (ITMSS) method. Then, according to the order of convergence analysis, the ITMSS method has a high order of convergence of at least 16. The results show that the ITMSS method converges faster than any single-step (IS2) procedure antecedently.

Furthermore, to evaluate the performance of the ITMSS method, we conducted a numerical experiment using 52 test examples, comparing it with IS2, ISS2 and IZSS2. We visualized the numerical results using the performance profile regarding the number of iterations and the largest final interval width. From the numerical results explained above, it is evident that the ITMSS method performs better than the IS2, ISS2, and IZSS2 methods, respectively. Although the proposed method is highly likely to win in Figure 1, this is not the case with the largest final interval width findings, as shown in Figure 2.

Figure 2 shows the comparison of all four methods in terms of the largest final interval width generated when reaching the stopping condition. From the graph, the ITMSS method has a high probability of winning for most of the scenarios, but the graph shows a less satisfactory outcome in some circumstances. In certain situations, the number of iterations for the ITMSS method is equivalent to the iterations by the IZSS2 method. Furthermore, there are also situations where even the ITMSS method has the fewest number of iterations, compared to the other three methods, but the value of its largest final interval width is similar. Next, we provide the details of the above situations in the tables.

From Table 2, we selected one of the test examples $p(x) = y^4 + \frac{40}{3}y^3 - 0.02y^2 - 0.4y$, which happened to have a less satisfactory outcome for ITMSS in terms of the number of

iteration generated. The table displays the interval width of every iteration by comparing all of the methods under consideration. At iteration k , the largest width of the final interval is highlighted in grey. As shown in Table 2, the polynomial required four iterations to complete for the IS2 method, whereas the ISS2 method only needed three iterations. However, both IZSS2 and ITMSS methods stopped at the second iteration. This table shows that the ITMSS method yields smaller intervals than other methods. For example, the width of $i = 2$ for the ITMSS method reached the stopping condition at $k = 1$; meanwhile, other i^{th} did not. All the i^{th} will continually run until the next iteration. Even though the ITMSS iterates twice as compared to the IZSS2 method, from the table, this method can obtain $[x_1^*, x_1^*]$ and $[x_4^*, x_4^*]$, which are essentially the roots for the first and fourth intervals, respectively. It can be seen that the largest final interval width for ITMSS is significantly smaller. This proved that the ITMSS method converges faster than other methods; it can also approximately reduce the final interval width for every iteration.

In Table 3, the polynomial $p(x) = 20000y^8 + 16080000y^7 + 551830000y^6 + 10534093200y^5 + 122028205260y^4 + 875779839648y^3 + 3789351757513y^2 + 8998687954893y + 8930298867308$ gives the final interval width generated for all methods as $8.88178419700125 \times 10^{-16}$. However, the ITMSS method has fulfilled the stopping conditions at $k = 1$. This observation, depending only on the largest final interval width value, does not significantly interpret the algorithm. However, indeed, the largest final interval width and the number of iterations are correlated to each other. From the overall view, the proposed algorithm can converge to the roots simultaneously and fulfill the inclusion theorem better than the other three methods. This means that the final width of the intervals generated is minimal. Hence, the ITMSS algorithm almost always converges to zero, irrespective of the initial estimates. From an overall view, imposing the value $g(x)$ and midpoint at every inner loop reduces the number of iterations and lessens the final interval width upon convergence. That is, the proposed method ensures that zero is contained within a suitably narrow final interval.

Table 2. Interval width of every iteration for polynomial $p(x) = y^4 + \frac{40}{3}y^3 - 0.02y^2 - 0.4y$.

k	i	Largest Interval Width in Every Iteration			
		IS2 Method	ISS2 Method	IZSS2 Method	ITMSS Method
1	1	$9.31439152433 \times 10^{-2}$	$2.38797944655 \times 10^{-4}$	$2.387979446556 \times 10^{-4}$	$2.48750330555 \times 10^{-7}$
	2	$1.08570553860 \times 10^{-2}$	$1.23524990965 \times 10^{-3}$	$1.234908302445 \times 10^{-3}$	$2.21779953451 \times 10^{-16}$
	3	$3.33618893631 \times 10^{-1}$	$4.98845687881 \times 10^{-2}$	$1.795679442520 \times 10^{-2}$	$6.61625643161 \times 10^{-10}$
	4	$1.95646289260 \times 10^{-2}$	$1.95646289260 \times 10^{-2}$	$1.582165843121 \times 10^{-3}$	$6.66966482043 \times 10^{-14}$
2	1	$4.32639438052 \times 10^{-6}$	$1.77635683940 \times 10^{-15}$	$1.776356839400 \times 10^{-16}$	0.0000000000000000
	2	$1.54184787449 \times 10^{-6}$	$3.65010213894 \times 10^{-16}$	$2.775557561562 \times 10^{-17}$	$2.21779953451 \times 10^{-17}$
	3	$1.04796990859 \times 10^{-3}$	$5.58678381334 \times 10^{-11}$	$2.775557561562 \times 10^{-17}$	$2.77555756156 \times 10^{-17}$
	4	$8.47766782607 \times 10^{-8}$	$2.13209447319 \times 10^{-9}$	$2.775557561562 \times 10^{-17}$	0.0000000000000000
3	1	$1.77635683940 \times 10^{-15}$	$1.77635683940 \times 10^{-16}$	Already Converge	Already Converge
	2	$1.52727551189 \times 10^{-17}$	$3.65010213894 \times 10^{-16}$		
	3	$1.70696790036 \times 10^{-14}$	$2.77555756156 \times 10^{-17}$		
	4	$2.77555756156 \times 10^{-17}$	$2.77555756156 \times 10^{-17}$		
4	1	$1.77635683940 \times 10^{-16}$	Already Converge	Already Converge	Already Converge
	2	$1.52727551189 \times 10^{-17}$			
	3	$2.77555756156 \times 10^{-17}$			
	4	$2.77555756156 \times 10^{-17}$			

Table 3. Interval width of every iteration for polynomial $p(x) = 20000y^8 + 16080000y^7 + 551830000y^6 + 10534093200y^5 + 122028205260y^4 + 875779839648y^3 + 3789351757513y^2 + 8998687954893y + 8930298867308$.

k	i	Largest Interval Width in Every Iteration			
		IS2 Method	ISS2 Method	IZSS2 Method	ITMSS Method
1	1	$8.790842687792 \times 10^{-1}$	$6.848788945959 \times 10^{-3}$	$6.848788945959 \times 10^{-3}$	$1.684182283412 \times 10^{-6}$
	2	$7.416880512258 \times 10^{-2}$	$2.425248230748 \times 10^{-2}$	$1.741756142319 \times 10^{-4}$	$7.105427357601 \times 10^{-15}$
	3	$9.554594425101 \times 10^{-3}$	$1.499200540824 \times 10^{-3}$	$1.702508947332 \times 10^{-5}$	$3.552713678800 \times 10^{-15}$
	4	$3.380631011767 \times 10^{-2}$	$1.910729522056 \times 10^{-3}$	$3.766384543979 \times 10^{-5}$	$3.552713678800 \times 10^{-15}$
	5	$3.691891521136 \times 10^{-2}$	$1.304172267009 \times 10^{-3}$	$3.477536074925 \times 10^{-5}$	$3.552713678800 \times 10^{-15}$
	6	$1.384620531206 \times 10^{-2}$	$6.131183766120 \times 10^{-4}$	$5.649886381409 \times 10^{-5}$	$1.776356839400 \times 10^{-15}$
	7	$1.554485059640 \times 10^{-1}$	$4.396253377089 \times 10^{-3}$	$7.726031239547 \times 10^{-4}$	$1.776356839400 \times 10^{-15}$
	8	$7.723378231853 \times 10^{-3}$	$7.723378231853 \times 10^{-3}$	$4.005897095238 \times 10^{-5}$	$8.881784197001 \times 10^{-16}$
2	1	$1.045554681976 \times 10^{-3}$	$7.105427357601 \times 10^{-15}$	$3.552713678800 \times 10^{-15}$	0.0000000000000000
	2	$7.953034852903 \times 10^{-7}$	$7.993605777301 \times 10^{-13}$	$1.776356839400 \times 10^{-15}$	0.0000000000000000
	3	$1.077414779615 \times 10^{-8}$	$5.329070518200 \times 10^{-15}$	$1.776356839400 \times 10^{-15}$	0.0000000000000000
	4	$3.946566735635 \times 10^{-7}$	$3.552713678800 \times 10^{-15}$	$1.776356839400 \times 10^{-15}$	0.0000000000000000
	5	$6.842841298038 \times 10^{-7}$	$3.552713678800 \times 10^{-15}$	$1.776356839400 \times 10^{-15}$	0.0000000000000000
	6	$3.241325474689 \times 10^{-7}$	$1.776356839400 \times 10^{-15}$	$8.881784197001 \times 10^{-16}$	0.0000000000000000
	7	$3.632269596209 \times 10^{-7}$	$1.776356839400 \times 10^{-15}$	$8.881784197001 \times 10^{-16}$	0.0000000000000000
	8	$8.304468224196 \times 10^{-14}$	$8.881784197001 \times 10^{-16}$	$4.440892098500 \times 10^{-16}$	$8.881784197001 \times 10^{-16}$
3	1	$7.105427357601 \times 10^{-15}$	0.0000000000000000	0.0000000000000000	
	2	$3.552713678800 \times 10^{-15}$	$1.776356839400 \times 10^{-15}$	0.0000000000000000	
	3	$1.776356839400 \times 10^{-15}$	0.0000000000000000	0.0000000000000000	
	4	$3.552713678800 \times 10^{-15}$	0.0000000000000000	$1.776356839400 \times 10^{-16}$	
	5	$3.552713678800 \times 10^{-15}$	0.0000000000000000	0.0000000000000000	
	6	$1.776356839400 \times 10^{-15}$	0.0000000000000000	$8.881784197001 \times 10^{-16}$	
	7	$8.881784197001 \times 10^{-16}$	0.0000000000000000	$8.881784197001 \times 10^{-16}$	
	8	$4.440892098500 \times 10^{-16}$	$8.881784197001 \times 10^{-16}$	$4.440892098500 \times 10^{-16}$	Already Converge
4	1	0.0000000000000000	0.0000000000000000		
	2	0.0000000000000000	0.0000000000000000		
	3	0.0000000000000000	0.0000000000000000		
	4	0.0000000000000000	0.0000000000000000		
	5	0.0000000000000000	0.0000000000000000	Already Converge	
	6	0.0000000000000000	0.0000000000000000		
	7	$8.881784197001 \times 10^{-16}$	0.0000000000000000		
	8	$4.440892098500 \times 10^{-16}$	$8.881784197001 \times 10^{-16}$		

5. Conclusions

In this study, we investigated the interval iterative methods for the inclusion of polynomial zeros specifically for solving simple roots problems simultaneously. We provide the numerical results, using a performance profile to validate the efficiency of all four methods: IS2, ISS2, IZSS2 and ITMSS. Theoretically, the proposed ITMSS method has an R-order of convergence of 16, which means it can bound the zeros rigorously at a superior rate. The numerical results indicate that the ITMSS method surpassed the other three methods by fine-tuning the midpoint and decreasing the final interval width with fewer iterations. However, it is suggested that further investigations can be conducted for polynomials with complex roots or with complex coefficients while solving the complexity of the computational CPU time running by implementing the interval arithmetic technique.

Author Contributions: Conceptualization, N.R.S. and C.Y.C.; methodology, N.R.S. and C.Y.C.; formal analysis, N.R.S. and Z.M.; software, N.R.S. and Z.M.; writing—original draft, N.R.S.; writing—review and editing, N.R.S., C.Y.C. and S.H.S.; visualization, N.R.S. and C.Y.C.; supervision, C.Y.C.; validation, C.Y.C. and S.H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by UNIVERSITI PUTRA MALAYSIA grant number GP-IPM/2021/9699800.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: No potential conflict of interest was reported by the authors.

Abbreviations

The following abbreviations are used in this manuscript:

IS2	Interval single-step method
ISS2	Interval symmetric single-step method
IZSS2	Interval zero-symmetric single-step method
ITMSS	Interval trio midpoint symmetric single-step method

References

- Ohura, R.; Minamoto, T. A blind digital image watermarking method based on the dyadic wavelet packet transform and fast interval arithmetic techniques. *Int. J. Wavelets Multiresolution Inf. Process.* **2015**, *13*, 1550040. [\[CrossRef\]](#)
- Rump, S.M.; Ogita, T.; Morikura, Y.; Oishi, S.I. Interval arithmetic with fixed rounding mode. *Nonlinear Theory and Its Applications. IEICE* **2016**, *7*, 362–373.
- Pereira, D.R.; Papa, J.P.; Saraiva, G.F.R.; Souza, G.M. Automatic classification of plant electrophysiological responses to environmental stimuli using machine learning and interval arithmetic. *Comput. Electron. Agric.* **2018**, *145*, 35–42. [\[CrossRef\]](#)
- Orozco-Gutierrez, M.L. An Interval-Arithmetic-Based Approach to the Parametric Identification of the Single-Diode Model of Photovoltaic Generators. *Energies* **2020**, *13*, 932. [\[CrossRef\]](#)
- Pan, W.; Feng, L.; Zhang, L.; Cai, L.; Shen, C. Time-series interval prediction under uncertainty using modified double multiplicative neuron network. *Expert Syst. Appl.* **2021**, *184*, 115478. [\[CrossRef\]](#)
- Weierstrass, K. Neuer Beweis des Satzes, dass jede ganze rationale Funktion einer Veränderlichen dargestellt werden kann als ein Product aus lineare Funktionen derselben Veränderlichen. *Gesammelte Werke* **1967**, *3*, 251–269.
- Proinov, P.D.; Cholakov, S.I. Semilocal Convergence of Chebyshev-like Root-finding Method for Simultaneous Approximation of Polynomial Zeros. *Appl. Math. Comput.* **2014**, *236*, 669–682. [\[CrossRef\]](#)
- Proinov, P.D.; Petkova, M.D. Convergence of The Two-point Weierstrass Root-finding Method. *Jpn. J. Ind. Appl. Math.* **2014**, *31*, 279–292. [\[CrossRef\]](#)
- Proinov, P.D.; Vasileva, M.T. On a Family of Weierstrass-type Root-finding Methods with Accelerated Convergence. *Appl. Math. Comput.* **2014**, *273*, 957–968. [\[CrossRef\]](#)
- Proinov, P.D.; Ivanov, S.I. On The Convergence of Halley's Method for Simultaneous Computation of Polynomial Zeros. *J. Numer. Math.* **2015**, *23*, 379–394. [\[CrossRef\]](#)
- Proinov, P.D.; Vasileva, M.T. On The Convergence of High-order Ehrlich-type Iterative Methods for Approximating All Zeros of A Polynomial Simultaneously. *J. Inequalities Appl.* **2015**, *2015*, 336. [\[CrossRef\]](#)
- Proinov, P.D. On The Local Convergence of Ehrlich Method for Numerical Computation of Polynomial Zeros. *Calcolo* **2016**, *253*, 413–426. [\[CrossRef\]](#)
- Proinov, P.D. Relationships Between Different Types of Initial Conditions for Simultaneous Root Finding Methods. *Appl. Math. Lett.* **2016**, *52*, 102–111. [\[CrossRef\]](#)
- Proinov, P.D.; Petkova, M.D. A New Semilocal Convergence Theorem for the Weierstrass Method for Finding Zeros of A Polynomial Simultaneously. *J. Complex.* **2014**, *30*, 366–380. [\[CrossRef\]](#)
- Cholakov, S.I.; Vasileva, M.T. A Convergence Analysis of A Fourth-order Method for Computing All Zeros of A Polynomial Simultaneously. *J. Comput. Appl. Math.* **2017**, *321*, 270–283. [\[CrossRef\]](#)
- Kyncheva, V.K.; Yotov, V.V.; Ivanov, S.I. Convergence of Newton, Halley and Chebyshev Iterative Methods as Methods for Simultaneous Determination of Multiple Polynomial Zeros. *J. Appl. Numer. Math.* **2017**, *112*, 146–154. [\[CrossRef\]](#)
- Proinov, P.D.; Ivanov, S.I. Convergence Analysis of Sakurai–Torii–Sugiura Iterative Method for Simultaneous Approximation of Polynomial Zeros. *J. Comput. Appl. Math.* **2019**, *357*, 56–70. [\[CrossRef\]](#)
- Gargantini, I.; Henrici, P. Circular Arithmetic and The Determination of Polynomial Zeros. *Numer. Math.* **1971**, *18*, 305–320. [\[CrossRef\]](#)
- Petković, M.S. On an iterative method for simultaneous inclusion of polynomial complex zeros. *J. Comput. Appl. Math.* **1982**, *8*, 51–56. [\[CrossRef\]](#)
- Monsi, M.; Wolfe, M.A. Interval Versions of Some Procedures for The Simultaneous Estimation of Complex Polynomial Zeros. *Appl. Math. Comput.* **1988**, *28*, 191–209. [\[CrossRef\]](#)
- Alefeld, G.; Herzberger, J. On the Convergence Speed of Some Algorithms for The Simultaneous Approximation of Polynomial Roots. *SIAM J. Numer. Anal.* **1974**, *11*, 237–243. [\[CrossRef\]](#)

22. Moore, R.E. *Methods and Applications of Interval Analysis*, 1st ed.; SIAM: Philadelphia, PA, USA, 1979.
23. Alefeld, G.; Herzberger, J. *Introduction to Interval Computations*, 1st ed.; Academic Press: New York, NY, USA, 1983.
24. Salim, N.R. Convergence of Interval Symmetric Single-step Method for Simultaneous Inclusion of Real Polynomial Zeros. Ph.D. Thesis, Universiti Putra Malaysia, Seri Kembangan, Malaysia, 2012.
25. Salim, N.R.; Monsi, M.; Hassan, M.A.; Leong, W.J. On The Convergence Rate of Symmetric Single-step Method ISS for Simultaneous Bounding Polynomial Zeros. *Appl. Math. Sci.* **2011**, *5*, 3731–3741.
26. Jamaludin, N.; Monsi, M.; Hassan, N. The Performance of The Interval Midpoint Zoro Symmetric Single-step (IMZSS2-5D) Procedure to Converge Simultaneously to The Zeros. In *AIP Conference Proceedings, Proceeding of The International Conference on Mathematics, Engineering and Industrial Applications 2018 (ICoMEIA 2018), Kuala Lumpur, Malaysia, 24–26 July 2018*; Zin, S.M., Abdullah, N., Khazali, K.A.M., Roslan, N., Rusdi, N.A., Saad, R.M., Yazid, N.M., Zain, N.A.M., Eds.; AIP Publishing LLC: Melville, NY, USA, 2018; Volume 2013, p. 020033.
27. Durand, E. *Solutions numériques des Équations algébriques: Systèmes de Plusieurs Équations*; Masson: Paris, France, 1960; Volume 2.
28. Kerner, I.O. Ein gesamtschrittverfahren zur berechnung der nullstellen von polynomen [A complete procedure for calculating the zeros of polynomials]. *Numer. Mathl Sci.* **2015**, *8*, 290–294. [\[CrossRef\]](#)
29. Rusli, S.F.; Monsi, M.; Hassan, M.A.; Leong, W.J. On the interval zoro symmetric single-step procedure for simultaneous finding of real polynomial zeros. *Appl. Math. Sci.* **2011**, *5*, 3693–3706.
30. Chen, C.Y.; Ghazali, A.H.; Leong, W.J. Scaled parallel iterative method for finding real roots of nonlinear equations. *Optimization* **2021**, 1–17. [\[CrossRef\]](#)
31. Ortega, J.M.; Rheinboldt, W.C. Numerical Solution of Nonlinear Problems: Studies in Numerical Analysis 2. *Symp. Spons. Nav. Res.* **1970**, *2*, 122–143.
32. Salim, N.R.; Monsi, M.; Hassan, N. On The Performances of IMZSS2 Method for Bounding Polynomial Zeros Simultaneously. In *Proceedings of the 7th International Conference on Research and Education in Mathematics (ICREM7), Kuala Lumpur, Malaysia, 25–27 August 2015*; Majid, Z.A., Salim, N.R., Laham, M.F., Gopal, K.; Phang, P.S., Mahad, Z., Eds.; IEEE: Piscataway, NJ, USA, 2015; pp. 5–9.
33. Rump, S.M. INTLAB — INTerval LABoratory. In *Developments in Reliable Computing*; Csendes, T., Ed.; Springer: Dordrecht, The Netherlands, 1999; pp. 77–104.
34. Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program.* **2002**, *91*, 201–213. [\[CrossRef\]](#)