Menemui Matematik (Discovering Mathematics)
Vol. 41, No. 1: 1-12 (2019)

Simultaneous Procedures with Newton Correction for Finding Real Zeros of Polynomial

Abdul Hakim Ghazali[1, a)], Chuei Yee Chen[1, 2, b)], Wah June Leong[1, 2, c)], Siti Mahani Marjugi[1,d)] and Nor Aliza Abd Rahmin[1, e)]

[1]*Department of Mathematics, Faculty of Science, Universiti Putra Malaysia,*
43400 UPM Serdang, Selangor, Malaysia.
[2]*Institute for Mathematical Research, Universiti Putra Malaysia,*
43400 UPM Serdang, Selangor, Malaysia.

[a)] akimghazali92@yahoo.com, [b)] Corresponding author: cychen@upm.edu.m, [c)] leongwj@upm.edu.my
[d)] smahani@upm.edu.my, [e)] aliza@upm.edu.my

ABSTRACT

Interval Single Step (IS1) procedure is an interval iterative procedure for finding real zeros of polynomial simultaneously. They have since been extended to Interval Symmetric Single Step (ISS1) and Interval Double Symmetric Single Step (IDSS1). In this paper, we propose the inclusion of Newton Correction to the procedures, of which we name as Interval Single Step with Newton Correction (IS1-N), Interval Symmetric Single Step with Newton Correction (ISS1-N) and Interval Double Symmetric Single Step with Newton Correction (IDSS1-N). Convergence analysis was done and the proposed procedures were tested on 120 polynomials. The efficiency of the modified methods were compared with existing procedures in terms of number of iterations and largest final interval width. The results show that the proposed procedures perform better than the original methods.

Keywords: Interval analysis, simultaneous method, single step, zeros of polynomial, Newton correction, performance profile

INTRODUCTION

Classically, all methods for finding zeros of polynomials only involve finding one root at a time and this can lead to increased rounding errors. On the contrary, the simultaneous method finds all the zeros of the polynomial at the same time. The advantage of the simultaneous method is that it nearly always converges to the zero no matter the value of the initial guess. This was verified by Semerdhiev (1994) where he found that for 4000 tested random polynomials, only 4 showed unsatisfactory results while the rests successfully converged. The first and simplest simultaneous method for finding real and simple zeros is Weierstrass, Durand and Kerner method (WDK) (see Weierstrass, 1891 and Weierstrass, 1903). This, together with rapid development of digital computers, leads to more modifications in ensuring the efficiency of the procedure (see also McNamee, 2007).

In 1959, Moore and Yang introduced interval analysis which is also known as interval arithmetic, interval computations or interval mathematics. The main significance of interval analysis is that more accurate results are guaranteed and they can be computed with finitely precise floating point operations. This idea was then applied on simultaneous method for finding real and simple zeros. Examples of early studies involving interval analysis on the simultaneous approach are by Gargantini and Henrici (1971) and Petkovic (1982). A more detailed literature of early iterative procedures involving interval can be found in McNamee (2007). Subsequently, Alefeld and Herzberger (1974) proposed the interval version of simultaneous iteration method known as Interval Single Step (IS1). Monsi (2011) then extended the IS1 procedure by introducing an extra step into the IS1 procedure. The procedure is known as Interval Symmetric Single Step (ISS1). This is then followed by Rusli et al. (2011) who extended ISS1 procedure to Interval Double Symmetric Single Step (IDSS1).

In this paper, we introduce Newton Correction into the IS1, ISS1 and IDSS1 procedures, of which we name our modification procedures as Interval Single Step with Newton Correction (IS1-N), Interval Symmetric Single Step with Newton Correction (ISS1-N) and Interval Double Symmetric Single Step with Newton Correction (IDSS1-N), respectively. Newton Correction is obtained from the part of the method by Schroder (1870). Similar to the WDK method, part of the method are used as correction on the method in Anourein (1977). The Newton Correction increases the convergence rate as the values of the polynomial and its derivative are evaluated at the midpoints, $x_i^{(k)}, i = 1,2,\dots,n$. This gives us the current approximations of the zeros. So the procedure with Newton Correction increases the speed of convergence and leads to faster convergence to the zeros. This correction is also used in several methods such as in Petkovic et al. (2003), Petkovic and Milosevic (2004) and Petkovic and Rancic (2006). While they focussed on the disk version, our method is applied on rectangles.

The paper is organized as follows. In Section 2, we provide the preliminaries of our study, focusing particularly on the method for estimating the polynomial zeros and interval iterative procedures. We present the algorithm of the modification procedures in Section 3. Subsequently, Section 4 will consist of a brief discussion on the convergence analysis of the procedures. In Section 5, we compare the efficiency of the new procedure with their original procedures in term of the number of iterations and largest final interval width by using performance profile. Finally, the conclusion of the paper is presented in Section 6.

PRELIMINARIES

The interval version of estimation of polynomial zeros goes as follow. Let $p\colon \mathbb{R} \to \mathbb{R}$ be a polynomial of degree n:

$$p(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i \qquad (1)$$

where $a_i \in \mathbb{R}, i = 1,\dots,n$ and $a_n \neq 0$. Suppose that p has n distinct zeros where $x_i^* \in \mathbb{R}, i = 1,\dots,n$. Also, note that $I(\mathbb{R})$ is the set of real intervals and $X_i^{(0)} \in I(\mathbb{R}), i = 1,\dots,n$ are such that

$$x_i^* \in X_i^{(0)}, \qquad i = 1,\dots,n, \qquad (2)$$

and the intervals are pairwise disjoint, that is

$$X_i^{(0)} \cap X_j^{(0)} = \emptyset, \qquad i,j = 1,\dots,n \text{ and } i \neq j. \qquad (3)$$

Assume that $a_n = 1$, so that

$$p(x) = \prod_{i=1}^{n} (x - x_i^*) \tag{4}$$

and from (4), we define

$$x_j^* = x - \frac{p(x)}{\prod_{j=1}^{n} \left(x - x_j^*\right)}. \tag{5}$$

Given the midpoints of the intervals $X_i^{(0)}, i = 1, \dots, n$ are

$$x_i^{(0)} = \text{midpoint}\left(X_i^{(0)}\right), \qquad i = 1, \dots, n. \tag{6}$$

Then by (2) and (3),

$$x_i^{(0)} \neq x_j^*, \qquad i, j = 1, \dots, n, j \neq i.$$

It follows from (5) that

$$x_j^* = x_i^{(0)} - \frac{p\left(x_i^{(0)}\right)}{\prod_{j=1}^{n} \left(x_i^{(0)} - x_j^*\right)}, \qquad i = 1, \dots, n. \tag{7}$$

Furthermore, by (3) and (6) with $x_i^{(0)} \notin X_j^{(0)}, i, j = 1, \dots, n, j \neq i$, we have

$$\prod_{j=1}^{n} \left(x_i^{(0)} - X_j^{(0)}\right) \neq 0, \qquad i = 1, \dots, n.$$

So, by (2) and (7) with the inclusion monotonicity of real interval, it follows that

$$x_i^* \in X_i^{(1)} = \left\{ x_i^{(0)} - \frac{p\left(x_i^{(0)}\right)}{\prod_{j=i}^{n} \left(x_i^{(0)} - X_j^{(0)}\right)} \right\} \cap X_i^{(0)}, \qquad i = 1, \dots, n.$$

The interval expression on the right is therefore a new interval $X_i^{(1)}$ for which

$$x_i^* \in X_i^{(1)} \subseteq X_i^{(0)}$$

holds. This relation gives rise to the following iteration,

$$X_i^{(k+1)} = \left\{ x_i^{(k)} - \frac{p\left(x_i^{(k)}\right)}{\prod_{j=i}^{n} \left(x_i^{(k)} - X_j^{(k)}\right)} \right\} \cap X_i^{(k)}, \qquad i = 1, \dots, n, \qquad k \geq 0.$$

This is Total Step (TS) procedure in terms of interval (see Alefeld and Herzberger, 1974 and Alefeld and Herzberger, 1983). Then they introduced IS1 (Alefeld and Herzberger, 1974b) procedure by introducing a new expression in the denominator of TS procedure as follows:

$$X_i^{(k,1)} = \left\{ x_i^{(k)} - \frac{p\left(x_i^{(k)}\right)}{\prod_{j=1}^{i-1} \left(x_i^{(k)} - X_j^{(k,1)}\right) \prod_{j=i+1}^{n} \left(x_i^{(k)} - X_j^{(k)}\right)} \right\} \cap X_i^{(k)},$$

for $i = 1, \dots, n,\ k \geq 0$. Its algorithm goes as follow:

Step 1: Given initial intervals $X_1^{(0)}, X_2^{(0)}, \dots, X_n^{(0)}$ and $X_i^{(0)} \cap X_j^{(0)} = \emptyset,\ i \neq j$.

Step 2: For $k \geq 0$, let $x_i^{(k)} = \text{midpoint}\left(X_i^{(k)}\right),\ i = 1, 2, \dots, n$.

For $i = 1,2, \dots, n$, evaluate

Step 3:

$$X_i^{(k,1)} = \left\{ x_i^{(k)} - \frac{p\left(x_i^{(k)}\right)}{\prod_{j=1}^{i-1}\left(x_i^{(k)} - X_j^{(k,1)}\right)\prod_{j=i+1}^{n}\left(x_i^{(k)} - X_j^{(k)}\right)} \right\} \cap X_i^{(k)}. \qquad (8)$$

Step 4: Set $X_i^{(k+1)} = X_i^{(k,1)}, i = 1, 2, \dots, n$.

Step 5: If the width of the interval is less than the stopping criteria $w\left(X_i^{k+1}\right) < \varepsilon$, then stop. Else set $k = k + 1$ and go to Step 2.

The procedure starts with initial intervals with each containing a zero. Step 3 of the algorithm, known as forward step, will generate intervals that are smaller than the initial intervals. The generated intervals decrease about half as we take the midpoint of the initial intervals in the iteration process. The zeros are guaranteed to be in the generated intervals as they intersect with the initial intervals, respectively. The process will repeat until the width of generated intervals achieve the stopping criteria.

The Interval Symmetric Single Step procedure (ISS1) is an extension of the IS1 procedure. ISS1 procedure has an extra step at each iteration known as the backward step:

$$X_i^{(k,2)} = \left\{ x_i^{(k)} - \frac{p\left(x_i^{(k)}\right)}{\prod_{j=1}^{i-1}\left(x_i^{(k)} - X_j^{(k,1)}\right)\prod_{j=i+1}^{n}\left(x_i^{(k)} - X_j^{(k,2)}\right)} \right\} \cap X_i^{(k,1)}, i = n, n-1, \dots, 1, \qquad (9)$$

which is performed after the forward step. ISS1 procedure is more efficient when compared to IS1 as ISS1 has one extra step at each iteration to generate smaller intervals. Meanwhile, the Interval Double Symmetric Single Step procedure (IDSS1) has a second forward step:

$$X_i^{(k,3)} = \left\{ x_i^{(k)} - \frac{p\left(x_i^{(k)}\right)}{\prod_{j=1}^{i-1}\left(x_i^{(k)} - X_j^{(k,3)}\right)\prod_{j=i+1}^{n}\left(x_i^{(k)} - X_j^{(k,2)}\right)} \right\} \cap X_i^{(k,2)}, \qquad (10)$$

for $i = 1, 2, \dots, n$. The word double in the name means there are two extra steps for the procedure, which are the backward step and the second forward step. The steps from each procedure (8), (9) and (10), respectively, will generate intervals that are smaller than the initial intervals. The zeros are guaranteed to be in the generated intervals as they intersect with the initial intervals, respectively. The process repeats until the width of generated intervals achieve the stopping criteria. With two extra steps in the IDSS1 procedure, the generated intervals will be the smallest among the three procedures as more calculations were made at each iteration. This is why IDSS1 is a more effective procedure as compared to IS1 and ISS1. In the next section, we show the modification on these three procedures.

ITERATIVE INTERVAL PROCEDURES WITH NEWTON CORRECTION

The Newton Correction is as follow:

$$N\left(x_i^{(k)}\right) = \begin{cases} \dfrac{p\left(\hat{x}_i^{(k)}\right)}{p'\left(x_i^{(k)}\right)} & \text{, if } p'\left(x_i^{(k)}\right) \neq 0, \\ 0 & \text{, if } p'\left(x_i^{(k)}\right) = 0 . \end{cases} \tag{11}$$

The Newton Correction increases the convergence rate as the correction already calculate values of p and p' at midpoints, $x_i^{(k)}, i = 1,2, \ldots, n$, which are the current approximations to the zeros. So the procedure with Newton Correction will increase the speed of convergence to the zeros. In our study, we include this Newton Correction into the second part of denominator of forward step as in Step 4 of Algorithm 1 below. The k^{th} interval sequence $X_i^{(k)}, i = 1, 2, \ldots, n$, generated by Interval Single Step with Newton Correction (IS1-N) procedure is given in Algorithm 1.

Algorithm 1 (IS1-N Procedure)

Step 1: Given initial intervals $X_1^{(0)}, X_2^{(0)}, \ldots, X_n^{(0)}$ and $X_i^{(0)} \cap X_j^{(0)} = \emptyset,\ i \neq j$.

Step 2: For $k \geq 0,\ x_i^{(k)} = \text{midpoint}\left(X_i^{(k)}\right),\ i = 1, 2, \ldots, n$.

Step 3: For $i = 1,2, \ldots, n$, evaluate $N\left(x_i^{(k)}\right)$.

Step 4: For $i = 1,2, \ldots, n$, evaluate

$$X_i^{(k,1)} = \left\{ x_i^{(k)} - \frac{p\left(x_i^{(k)}\right)}{\prod_{j=1}^{i-1}\left(x_i^{(k)} - X_j^{(k,1)}\right)\prod_{j=i+1}^{n}\left(x_i^{(k)} - X_j^{(k)} + N\left(x_i^{(k)}\right)\right)} \right\} \cap X_i^{(k)}. \tag{12}$$

Step 5: Set $X_i^{(k+1)} = X_i^{(k,1)}, i = 1, 2, \ldots, n$.

Step 6: If $w(X_i^{k+1}) < \varepsilon$, then stop. Else set $k = k + 1$ and go to Step 2.

The Interval Symmetric Single Step with Newton Correction (ISS1-N) procedure is an extension from the IS1-N procedure by applying the concept of the ISS1 procedure. This backward step generates smaller intervals as the calculation of each iteration is made twice. This leads to faster convergence. The k^{th} interval sequence $X_i^{(k)}, i = 1, 2, \ldots, n$, generated by ISS1-N procedure is in Algorithm 2.

Algorithm 2 (ISS1-N Procedure)

Step 1: Given initial intervals $X_1^{(0)}, X_2^{(0)}, \ldots, X_n^{(0)}$ and $X_i^{(0)} \cap X_j^{(0)} = \emptyset,\ i \neq j$.

Step 2: For $k \geq 0,\ x_i^{(k)} = \text{midpoint}\left(X_i^{(k)}\right),\ i = 1, 2, \ldots, n$.

Step 3: For $i = 1,2, \ldots, n$, evaluate $N\left(x_i^{(k)}\right)$.

Step 4: For $i = 1,2, \ldots, n$, evaluate forward step as in (12).

Step 5: For $i = n, n - 1, \ldots, 1$, evaluate backward step as in (9).

Step 6: Set $X_i^{(k+1)} = X_i^{(k,2)}, i = 1, 2, \ldots, n$.

Step 7: If $w(X_i^{k+1}) < \varepsilon$, then stop. Else set $k = k + 1$ and go to Step 2.

The Interval Double Symmetric Single Step with Newton Correction (IDSS1-N) procedure is an extension of ISS1-N with an extra step, second forward step (10) after the backward step. With the two extra steps in IDSS1-N procedure, it generates smaller intervals when compared to the ISS1-N procedure, hence leading to faster convergence and the procedure yields lesser number of iterations k. The k^{th} interval sequence $X_i^{(k)}, i = 1, 2, \dots, n$, generated by IDSS1-N procedure is given in Algorithm 3.

Algorithm 3 (IDSS1-N Procedure)

Step 1: Given initial intervals $X_1^{(0)}, X_2^{(0)}, \dots, X_n^{(0)}$ and $X_i^{(0)} \cap X_j^{(0)} = \emptyset,\ i \neq j$.
Step 2: For $k \geq 0,\ x_i^{(k)} = \text{midpoint}\left(X_i^{(k)}\right),\ i = 1, 2, \dots, n$.
Step 3: For $i = 1,2, \dots, n$, evaluate $N\left(x_i^{(k)}\right)$.
Step 4: For $i = 1,2, \dots, n$, evaluate forward step as in (12).
Step 5: For $i = n, n-1, \dots, 1$, evaluate backward step as in (9).
Step 6: For $i = 1, \dots, n$, evaluate second forward step as in (10).
Step 7: Set $X_i^{(k+1)} = X_i^{(k,3)}, i = 1, 2, \dots, n$.
Step 8: If $w(X_i^{k+1}) < \varepsilon$, then stop. Else set $k = k + 1$ and go to Step 2.

CONVERGENCE ANALYSIS

We now give the convergence analysis of the procedures, of which we show that the generated intervals always decrease towards the zeros.

Theorem 1

Let

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

be a polynomial with *n* simple roots, x_i^* and pairwise disjoint initial interval, $X_i^{(0)}$ with $x_i^* \in X_i^{(0)}, i = 1, \dots, n$. Further let $a_n = 1$. Then the sequence $\{X_i^{(k)}\}_{k=0}^{\infty}$ generated from IS1-N, ISS1-N and IDSS1-N procedures satisfy

$$x_i^* \in X_i^{(k)}, \qquad k \geq 0$$

and

$$X_i^{(0)} \supset X_i^{(1)} \supset X_i^{(2)} \supset \cdots$$

with $\lim_{k\to\infty} X_i^{(k)} = x_i^*$, or the sequence comes to rest at $[x_i^*, x_i^*]$ after a finite number of iteration, where $[x_i^*, x_i^*]$ is a zero in term of interval. Define $D_i = \left[d_{i\,\text{inf}}, d_{i\,\text{sup}}\right]$. If $0 \notin D_i \subset I(\mathbb{R})$ is such that $p'(x) \in D_i$ for all $x \in X_i^{(0)}$, then

$$w\left(X_i^{(k+1)}\right) \leq \frac{1}{2}\left(1 - \frac{d_{i\,\text{inf}}}{d_{i\,\text{sup}}}\right) w\left(X_i^{(k)}\right),$$

where $w\left(X_i^{(k)}\right) = w\left(\left[x_{i\,\text{inf}}^{(k)}, x_{i\,\text{sup}}^{(k)}\right]\right) = x_{i\,\text{sup}}^{(k)} - x_{i\,\text{inf}}^{(k)}$.

Proof:

We only show the first part of the theorem while the second part of the theorem follows from Alefeld and Herzberger (1983), Monsi (2011) and Rusli et al. (2011). Let

$$S_i^{(k)} = \prod_{j=1}^{i-1}\left(x_i^{(k)} - X_j^{(k+1)}\right)\prod_{j=i+1}^{n}\left(x_i^{(k)} - X_j^{(k)} + N\left(x_i^{(k)}\right)\right).$$

We estimate

$$d\left(X_i^{(k+1)}\right) \leq d\left\{x_i^{(k)} - \frac{p\left(x_i^{(k)}\right)}{S_i^{(k)}}\right\} \leq d\left(\frac{p\left(x_i^{(k)}\right)}{S_i^{(k)}}\right) = \left|p\left(x_i^{(k)}\right)\right| d\left(\frac{1}{S_i^{(k)}}\right). \quad (13)$$

Observe that

$$\left|p\left(x_i^{(k)}\right)\right| = \left|p\left(x_i^{(k)}\right) - p(x_i^*)\right| \quad (14)$$

since $p(x_i^*) = 0$. By letting

$$p'\left(x_i^{(k)}\right) = \frac{p\left(x_i^{(k)}\right) - p(x_i^*)}{x_i^{(k)} - x_i^*},$$

we now have

$$\left|p\left(x_i^{(k)}\right)\right| \leq \left|\left(x_i^{(k)} - x_i^*\right)p'\left(x_i^{(k)}\right)\right| = d\left(X_i^{(k)}\right)\left|p'\left(x_i^{(k)}\right)\right| \leq d\left(X_i^{(k)}\right)\left|p'\left(X_i^{(0)}\right)\right|,$$

and it follows from (13) that

$$d\left(X_i^{(k+1)}\right) \leq d\left(X_i^{(k)}\right)\left|p'\left(X_i^{(0)}\right)\right| d\left(\frac{1}{S_i^{(k)}}\right).$$

The rest of the proof follows from Alefeld and Herzberger (1983) that gives us

$$d\left(X_i^{(k+1)}\right) \leq d\left(X_i^{(k)}\right)$$

and it is valid that $X_i^{(k)} \subseteq X_i^{(0)}$. Therefore,

$$X_i^{(0)} \supset X_i^{(1)} \supset X_i^{(2)} \supset \cdots.$$

So, we conclude that $\lim_{k\to\infty} X_i^{(k)} = x_i^*$.

RESULTS AND DISCUSSION

The procedures were tested on 120 test problems. We compare the efficiency of the modified procedure with its original procedure in terms of number of iterations and largest final interval width. All algorithms were run using Matlab with Intlab V5.5 toolbox of Rump (1999) with stopping criteria of interval width $w\left(X_i^{k+1}\right) < 10^{-15}$. The computational results are presented using performance profile (Dolan and More, 2002).

Performance profile is an analytical tool with visualization that is used to interpret the result using benchmark experiment. It allows one to evaluate and compare the performance of a set of solvers on a given set of test, with respect to a chosen evaluation parameter. The performance profile is also able to indicate the most and the least efficient solvers.

Figure 1 shows the profiling graph of the modification procedures and their original procedures in terms of number of iteration. From the figure, it is obvious that the modification procedure IDSS1-N perform better than its original procedure IDSS1 since IDSS1-N has the highest probability of being the optimal solver compared to others. In other words, IDSS1-N procedure requires less iterations to converge to zero. However, the procedure ISS1-N shows less satisfactory results. Statistically, ISS1-N procedure can solve with the best result for about 32%, while for ISS1 about 34%. The difference is intangible.

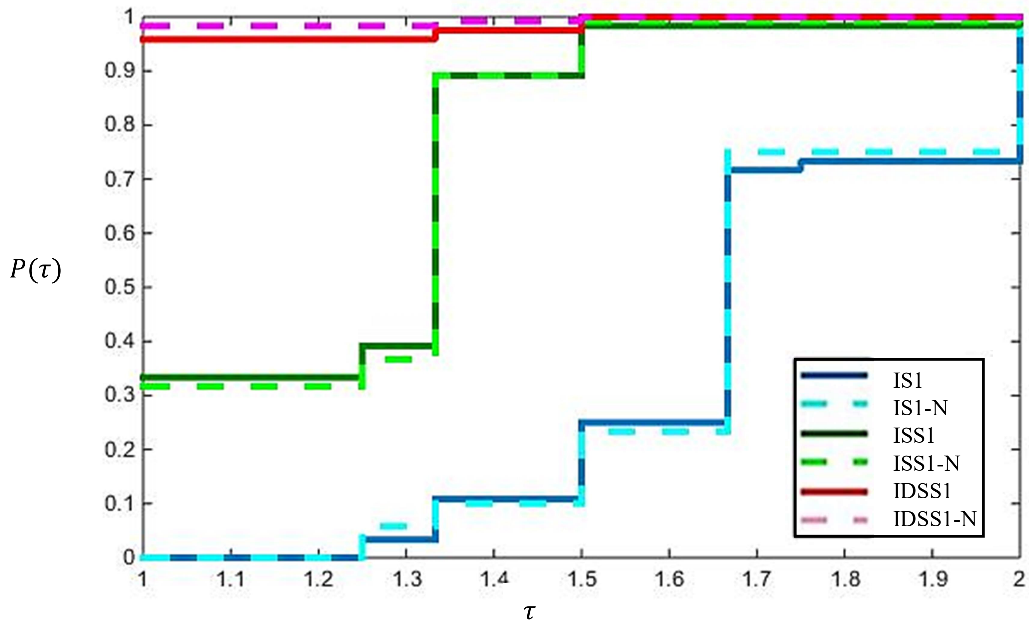

Figure 1: Performance Profile for Comparison in terms of Number of Iteration

As for IS1 and IS1-N, it is not straightforward to predict which is the better procedure since Figure 1 analyses the data from all the procedures. With the data being simulated in the same programming, it is difficult to compare the two solvers that have little differences in their results (Gould and Scott, 2016). Therefore we need to generate a performance profile for both procedures only, as shown in Figure 2. From the figure, it is apparent that IS1-N procedure perform better than IS1 procedure. Statistically, the probability that IS1-N can solve the polynomial with less number of iteration is about 0.875 and for IS1 about 0.840.

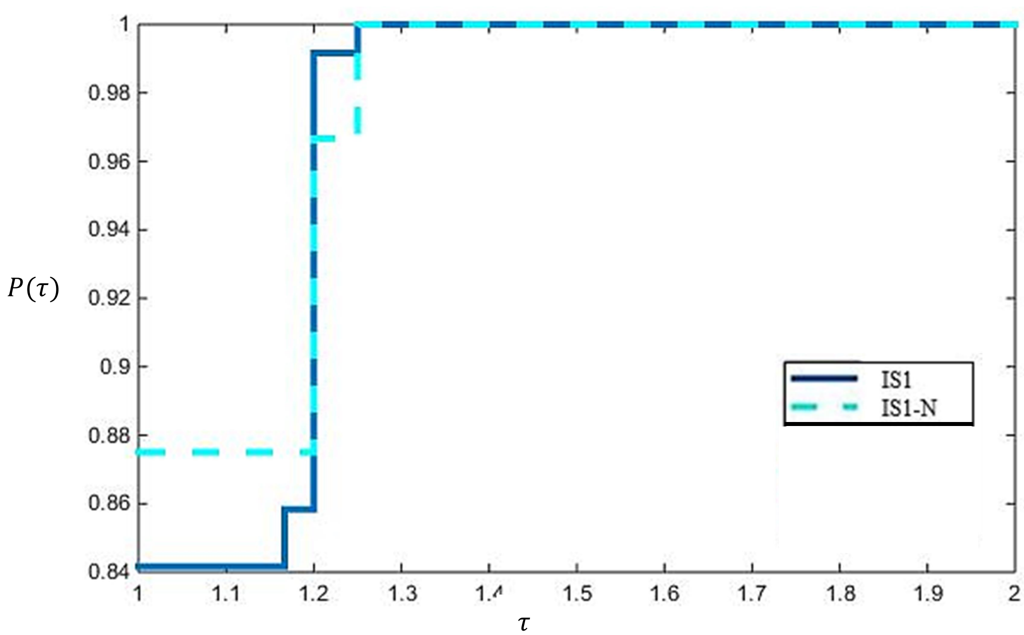

Figure 2: Performance Profile for Comparison in terms Number of Iteration for IS1 and IS1-N

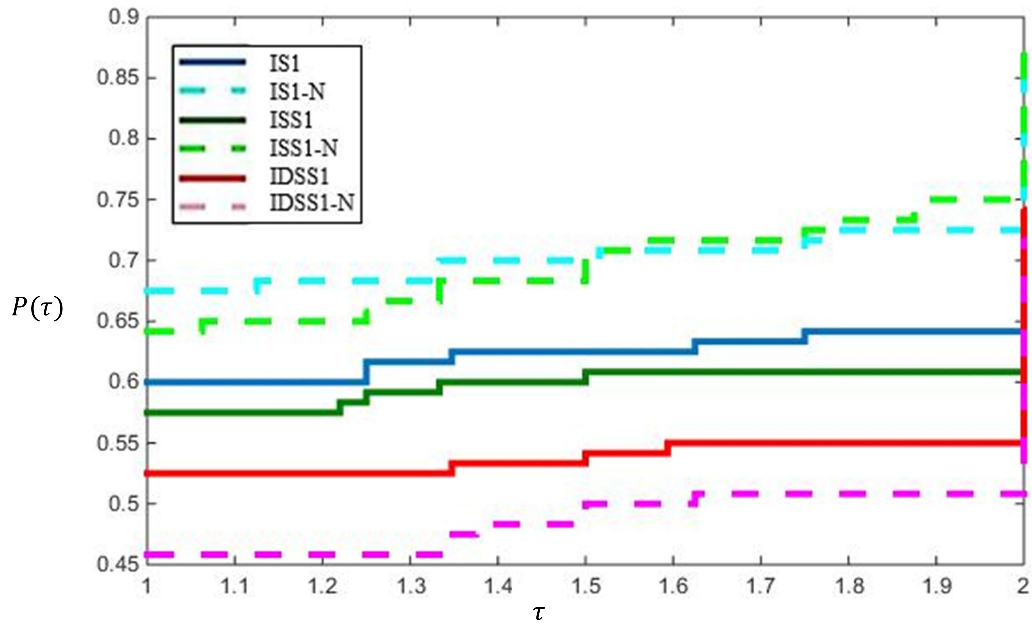

Figure 3: Performance Profile for Comparison in terms on Largest Final Interval Width

Next, Figure 3 presents the performance profile for the largest final interval width of the procedures. Though IS1-N and ISS1-N may not outperform their original procedures in terms of number of iterations, their performance is much better in terms of final interval width. This simply means that the width of the final interval after convergence for these two modification procedures are smaller as compared to their original procedures, respectively. In other words, IS1-N and ISS1-N procedures converges faster than their original procedures, respectively, even though the number of iterations are almost the same. However, IDSS1-N procedure shows the least satisfying result. The procedure has less probability of solving the zeros of the polynomial with the small final interval width. This procedure has higher value of final interval width as compared to its original procedure, IDSS1 procedure. However, note that IDSS1-N procedure is the most effective method in terms of number of iterations, as one can see from Figure 1. In some cases, the number of iteration may be lesser but the final interval widths is larger. We give two examples of polynomial that reflect this situation in Table 1 and Table 2.

Table 1: Interval width in every interval and iteration for polynomial 7*

k	i	Interval width in every interval and iteration	
		IDSS1	IDSS1-N
1	1	0.001724877177355	0.001159999981514
	2	0.002695201047809	0.002261232799590
	3	9.135187853126503e-05	7.392793855998114e-05

* $p(x) = x^3 - 3x + 1$ with initial intervals [-2.5,-1.1], [-1,0.9], [1.1,1.9]

2	1	1.865174681370263e-14	2.220446049250313e-15
	2	6.106226635438361e-16	2.220446049250313e-16
	3	2.220446049250313e-16	2.220446049250313e-16
3	1	2.220446049250313e-16	-
	2	6.106226635438361e-16	-
	3	2.220446049250313e-16	-

Table 1 presents the interval width in every interval, i and iteration, k for one of the tested polynomials. From the table, all interval width for IDSS1-N procedure has already converged at iteration $k = 2$. However, interval $i = 1$ for IDSS1 procedure at iteration $k = 2$ has not converged yet and still continue to run until it converges at the next iteration. The IDSS1 procedure then yields a smaller interval width at the next iteration, but IDSS1-N procedure has already converged and the calculation has stopped. That is why some of the final interval width for IDSS1 procedure can be smaller when compared to the modified procedure. Table 2 also shows the same situation happening in another tested polynomial. At $k = 3$, IDSS1-N has already converged but IDSS1 has not since interval $i = 1$ still does not satisfy the stopping criteria. The largest final interval width for IDSS1-N is at interval $i = 1$ while for IDSS1 is at interval $i = 7$ at the next iteration. If we compare each of the final interval i for both procedures, we notice that most width of the interval of the IDSS1-N are smaller compared to the IDSS1 procedure.

Table 2: Interval width in every interval and iteration for polynomial 40†

k	i	Interval width in every interval and iteration	
		IDSS1	IDSS1-N
1	1	1.148940650673406	1.182805209119608
	2	0.002139186578885	0.002138709250000
	3	0.474033673950179	0.478223081911198
	4	0.805774549993387	0.813706397073050
	5	0.832956783535940	0.850388317535850
	6	0.719417907840691	0.740747980803507
	7	1.518209408891533	1.554760048180345
	8	2.075052883444817	2.124631664509566
2	1	0.004539471521884	0.004420346507992
	2	2.000983600014550e-05	1.853746986046900e-05
	3	5.575996931592719e-04	4.894479972621380e-04
	4	9.632299688834695e-04	8.929302637096653e-04
	5	6.084534814165821e-04	5.527433974310547e-04

† $p(x) = x^8 + 26.8562x^7 + 165.507x^6 - 487.737x^5 - 4265.98x^4 + 5980.42x^3 + 25347.1x^2 - 38639.3x$ with initial intervals [2.9,4.9], [2.1,2.8], [0.8,2.0], [-1.8,0.7], [-5.8,-1.9], [-8.1,-5.9], [-13.8,-8.7] and [-22,-13.9]

	6	0.001718927605319	0.001792000071357
	7	3.853130611979339e-04	3.545209526691906e-04
	8	3.327292892585376e-05	2.619950278592853e-05
3	1	1.465494392505207e-14	4.884981308350689e-15
	2	4.440892098500626e-16	4.440892098500626e-16
	3	4.440892098500626e-16	2.220446049250313e-16
	4	2.707625519193097e-17	3.078964763269382e-18
	5	8.881784197001252e-16	8.881784197001252e-16
	6	8.881784197001252e-16	8.881784197001252e-16
	7	1.776356839400251e-15	1.776356839400251e-15
	8	1.776356839400251e-15	1.776356839400251e-15
4	1	4.440892098500626e-16	-
	2	4.440892098500626e-16	-
	3	4.440892098500626e-16	-
	4	2.707625519193097e-17	-
	5	8.881784197001252e-16	-
	6	8.881784197001252e-16	-
	7	1.776356839400251e-15	-
	8	1.776356839400251e-15	-

CONCLUSION

We have presented three modified procedures which we name as Interval Single Step with Newton Correction (IS1-N), Interval Symmetric Single Step with Newton Correction (ISS1-N) and Interval Double Symmetric Single Step with Newton Correction (IDSS1-N). From the results, the proposed procedures are more efficient as compared to their original procedures, respectively. Though the Newton correction does not bring much effect on the IS1 and ISS1 procedures in term of decreasing the number of iterations, it increases the efficiency in term of final interval width, i.e. IS1-N and ISS1-N converge to the zeros faster than their original procedure, respectively. Meanwhile the Newton correction contributes to decreasing the number of iterations of IDSS1 method but the largest final interval width for IDSS1-N may not be smaller due to its faster convergence, i.e. reaching stopping criteria with lesser number of iterations.

ACKNOWLEDGEMENT

This research is supported by Universiti Putra Malaysia under Fundamental Research Grant Scheme FRGS/1/2016/STG06/UPM/02/9.

REFERENCES

Alefeld, G., & Herzberger, J. (1974). Über Simultanverfahren zur Bestimmung reeller Polynomwurzeln. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, *54*(6), 413-420.

Alefeld, G., & Herzberger, J. (1983). *Introduction to interval computation*. Academic press.

Anourein, A. W. M. (1977). An improvement on two iteration methods for simultaneous determination of the zeros of a polynomial. *International Journal of Computer Mathematics*, *6*(3), 241-252.

Durand, E. (1960), *Solution Num´erique des ´ Equations Alg´ebriques, Vol. 1, Equation du Type F(x) = 0, Racines d'une Polynˆome*. Masson, Paris.

Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, *91*(2), 201-213.

Gargantini, I., & Henrici, P. (1971). Circular arithmetic and the determination of polynomial zeros. *Numerische Mathematik*, *18*(4), 305-320.

Gould, N., & Scott, J. (2016). A note on performance profiles for benchmarking software. *ACM Transactions on Mathematical Software (TOMS)*, *43*(2), 15.

Kerner, I. O. (1966). Ein gesamtschrittverfahren zur berechnung der nullstellen von polynomen. *Numerische Mathematik*, *8*(3), 290-294.

McNamee, J. M. (2007). *Numerical Methods for Zeros of Polynomials Part 1.* First Edition, United Kingdom: Elsevier Publishing Company.

Moore, R. E., & Yang, C. T. (1959). Interval analysis I. *Technical Document LMSD-285875, Lockheed Missiles and Space Division, Sunnyvale, CA, USA*.

Monsi, M. (2011). The interval symmetric single-step ISS1 procedure for simultaneously bounding simple polynomial zeros. *Malaysian Journal of Mathematical Sciences*, *5*(2), 211-227.

Petković, M. S. (1982). On an iterative method for simultaneous inclusion of polynomial complex zeros. *Journal of Computational and Applied Mathematics*, *8*(1), 51-56.

Petkovic, M. S., & Miloševic, D. M. (2004). Laguerre-like methods with corrections for the inclusion of polynomial zeros. *Novi Sad J. Math*, *34*(1), 135-156.

Petković, M. S., Petković, L. D., & Rančić, L. (2003). Higher-order simultaneous methods for the determination of polynomial multiple zeros. *International journal of computer mathematics*, *80*(11), 1407-1427.

Petkovic, M. S., & Rancic, L. (2006). A family of root-finding methods with accelerated convergence. *Computers and Mathematics with Applications*, *51*(6-7), 999-1010.

Rump, S. M. (1999). INTLAB—interval laboratory. *In Developments in reliable computing* (pp. 77-104). Springer, Dordrecht.

Rusli, S. F. M., Monsi, M., Hassan, M. A., & Leong, W. J. (2011). On the interval zoro symmetric single-step procedure for simultaneous finding of polynomial zeros. *Applied Mathematical Sciences*, *5*(75), 3693-3706.

Schröder, E. (1870). Über unendlich viele Algorithmen zur Auflösung der Gleichungen. *Mathematische Annalen*, *2*(2), 317-365.

Weierstrass, K. (1903), Neuer Beweis des Satzes, dass jede ganze rationale Function einer Veranderlichen dargestellt werden kann als ein Product aus linearen Functionen derselben Veranderlichen. *Ges, Werke* 3, 251-269.

Weierstrass, K. (1891), Neuer Beweis des Satzes, dass jede ganze rationale Function einer Ver¨anderlichen dargestellt werden kann als ein Product aus linearen Functionen derselben Ver¨anderlichen. *Sitzungsber. Konigl Akad. Wiss. Berlin*, 1085-1101.