On Computational Efficiency of the Iterative Methods for the Simultaneous Approximation of Polynomial Zeros

G. V. MILOVANOVIĆ and M. S. PETKOVIĆ
University of Niš

A measure of efficiency of simultaneous methods for determination of polynomial zeros, defined by the *coefficient of efficiency*, is considered. This coefficient takes into consideration (1) the R-order of convergence in the sense of the definition introduced by Ortega and Rheinboldt (*Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, 1970) and (2) the number of basic arithmetic operations per iteration, taken with certain weights depending on a processor time. The introduced definition of computational efficiency was used for comparison of the simultaneous methods with various structures.

Categories and Subject Descriptors: G.1.5 [**Numerical Analysis**]: Roots of Nonlinear Equations—*iterative methods*

General Terms: Measurement, Performance

Additional Key Words and Phrases: Algebraic equation, computational efficiency, polynomial zeros, simultaneous methods

1. INTRODUCTION

Over the last 20 years many iterative formulas for the simultaneous determination of all zeros of a polynomial have been established. In practice, it is important to know the characteristics of these methods relative to the number of numerical operations in finding polynomial zeros with the wanted accuracy, convergence speed, processor time of a computer, taking possession of storage space at a computer, etc. An estimation of efficiency of simultaneous methods, taking into consideration the above points, has not been treated in the literature yet.

The purpose of this paper is to introduce a measure of efficiency of simultaneous processes, defined by the *coefficient of efficiency*. This coefficient takes into consideration (1) the R-order of convergence (in the sense of the definition introduced by Ortega and Rheinboldt [14]) and (2) the number of basic arithmetic operations per iteration, taken with certain weights depending on processor time. This definition of efficiency enables the simultaneous methods to be compared

Authors' address: Faculty of Electronic Engineering, Department of Mathematics, University of Niš, P.O. Box 73, 18 000 Niš, Yugoslavia.

with various structures (e.g., those with derivatives or without them, in serial or parallel fashion).

Most of the known simultaneous methods are analyzed in this paper. The comparison of these methods in view of efficiency is performed for various values of the polynomial degree. For simplicity, only the case of real and simple zeros is considered. This analysis can be easily expanded to the complex case.

2. COMPUTATIONAL EFFICIENCY

Let $\{\mathbf{x}^m\}$ be an iterative sequence generated by some iterative function (shorter IF) solving a nonlinear (algebraic or transcendental) equation $f(x) = 0$. A measure of the informational usage by an IF and a measure of the efficiency of the IF are required. Taking the informational usage d of an IF as the number of new pieces of information required per iteration, Traub [17, p. 11] introduced the following definition:

The informational efficiency E is the convergence order of IF r, divided by the informational usage d; that is,

$$E = \frac{r}{d}.$$

The informational usage d is the total number of new function evaluations (the values of f and its derivatives) per iteration.

Ostrowski [15, p. 20] gave an alternative definition of efficiency, the *efficiency index*,

$$^{*}E = r^{1/d}.$$

The concept of informational efficiency does not take into account the cost of evaluating the values of function f and its derivatives. The *computational efficiency* of an iterative function ϕ relative to f, which does take these costs into consideration, was introduced by Traub [17, pp. 260–264]:

$$E(\phi, f) = r^{1/\Theta},$$

where $\Theta = \sum \theta_j$ and θ_j is the cost of evaluating $f^{(j)}$. If the informational usage of ϕ is d, and if θ_j is independent of j, then $E(\phi, f)$ is independent of f and reduces to

$$^{*}E = E(\phi) = r^{1/d}.$$

Now, let f be a monic algebraic polynomial P of degree n with real simple zeros $\xi_1, \ldots, \xi_n$; that is,

$$\mathrm{P}(x) = x^n + a_1 x^{n-1} + \cdots + a_{n-1}x + a_n = \prod_{i=1}^{n} (x - \xi_i) \quad (a_i \in R).$$

Applying the simultaneous methods for determination of all zeros of the polynomial P, one forms n sequences $\{x_1^m\}, \ldots, \{x_n^m\}(m = 1, 2, \ldots)$ starting with a reasonably good initial approximation $x_1^0, \ldots, x_n^0$. The previous definitions of efficiency refer to one sequence only so that they cannot be applied for the simultaneous methods where n mutually dependent sequences are produced. Furthermore, in generating these sequences by some iterative formula, besides the evaluations of polynomial and (eventually) its derivatives, the necessity for evaluation of some arithmetic expressions (sums, products, and so on), depending on the approximations of polynomial zeros (the terms of sequences $\{x_k^m\}$, $k = 1, \ldots, n$), appears. In order to define the cost of iteration, a heterogenous (intermixed) structure imposes the necessity of introducing the total number of basic arithmetic operations (addition, subtraction, multiplication, division) for all zeros per iteration.

In practical application of simultaneous methods, the Gauss-Seidel approach (which uses the approximations to the zeros immediately as they become available) is the most frequently used. Besides the accelerated convergence attained without additional evaluations, this procedure is also favorable relative to the occupation of storage space at a digital computer. The analysis of the convergence order of these methods in serial fashion is provided by the concept of the R-order of convergence. Since the R-order of convergence depends on the polynomial degree n, we will denote it with $r(n)$ in the sequel.

It is obvious that any simultaneous method is more efficient if its R-order of convergence is greater and the total number of basic arithmetic operations per iteration is smaller. Defining a coefficient of efficiency, it is also necessary to take into consideration the processor time needed for execution of the mentioned operations. Because of that, we will correspond to each operation the *weight* that is (1) proportional to the number of elementary steps (period clocks) necessary in the execution of this operation in the arithmetic units of the computer and (2) normalized in reference to the addition. These weights will be denoted by w_A, w_S, w_M, and w_D for addition, subtraction, multiplication, and division, respectively, setting $w_A = 1$ because of the normalization.

Let us analyze now the number of necessary arithmetic operations per iteration. We presume that the Horner scheme is used for the evaluation of the given polynomial and its derivatives (if they appear).

In evaluating the polynomial values at n point approximations to the zeros, taking into account the weights of the basic operations, the corresponding cost of evaluation of the polynomial P can be defined as follows:

$$\mathrm{G}(n) = w_A\overline{\mathrm{A}}(n) + w_M\overline{\mathrm{M}}(n).$$

Here $\overline{\mathrm{A}}(n)$ and $\overline{\mathrm{M}}(n)$ denote the number of additions and multiplications, respectively. It is well known that

$$\overline{\mathrm{A}}(n) = n^2, \qquad \overline{\mathrm{M}}(n) = n^2$$

for a real polynomial of degree n.

The number of all remaining operations, which are necessary in realization of one iteration, will be denoted by

$\mathrm{A}(n)$ (additions),
$\mathrm{S}(n)$ (subtractions),
$\mathrm{M}(n)$ (multiplications), and
$\mathrm{D}(n)$ (divisions).

The quantities $A(n)$ and $M(n)$ include the operations necessary for evaluation of the derivatives of the polynomial (by the Horner scheme), when they appear.

The total cost of the evaluation (for all zeros) per iteration is equal to

$$T(n) = G(n) + w_A A(n) + w_S S(n) + w_M M(n) + w_D D(n).$$

It is convenient to introduce the normalized cost of evaluation

$$\Theta(n) = \frac{T(n)}{G(n)};$$

that is,

$$\Theta(n) = 1 + \frac{w_A A(n) + w_S S(n) + w_M M(n) + w_D D(n)}{G(n)}.$$

Finally, according to the previous consideration, we may define the coefficient of the efficiency of the simultaneous iterative process (shorter SIP) for finding polynomial zeros.

Definition. If $r(n)$ is the R-order of convergence of the simultaneous iterative process SIP and $\Theta(n)$ is the normalized cost of evaluation, then

$$(E_1)\quad E(\mathrm{SIP}, n) = \frac{r(n)}{\Theta(n)}$$

will be called the coefficient of the efficiency of SIP.

An alternative definition of the coefficient of efficiency, analogous to the Ostrowski efficiency index, can be introduced by

$$(E_2)\quad {}^*E(\mathrm{SIP}, n) = r(n)^{1/\Theta(n)}.$$

But the formula (E_2) is more complicated for computation and shows greater disagreement with the experimental results (measuring CPU time in realization of an iterative process on a computer) compared to the above definition (E_1). Therefore, in the sequel we will use the formula (E_1).

Note that the definitions (E_1) and (E_2) can also be applied in the case of complex zeros of the polynomial.

3. LIST OF METHODS

In this section we will give a review of the most frequently used simultaneous methods for polynomial zeros. First, let us introduce some notations:

1° The approximations of the zeros $x_1^m, \ldots, x_n^m$ in the mth iteration will be briefly denoted with $x_1, \ldots, x_n$, and new approximations $x_1^{m+1}, \ldots, x_n^{m+1}$, obtained by some simultaneous methods, by $\hat{x}_1, \ldots, \hat{x}_n$, respectively;

2° $Q(x) = (x - x_1)(x - x_2) \cdots (x - x_n)$;

3° $W_k = P(x_k)/Q'(x_k)$ (Weierstrass's correction);

4° $N_k = P(x_k)/P'(x_k)$ (Newton's correction).

3.1 Iterative Formulas without Derivatives

One of the most popular formulas for the simultaneous approximations of the polynomial zeros is as follows:

$$\hat{x}_k = x_k - \frac{\mathrm{P}(x_k)}{\prod_{i=1,i\neq k}^{n} (x_k - x_i)} \quad (k = 1, \ldots, n). \tag{1}$$

Formula (1) is the classical result introduced by Weierstrass [18, p. 258] in 1891, in connection with a proof of the fundamental theorem of algebra. However, this formula found its application for the simultaneous determination of polynomial zeros much later (Dočev [5], Durand [6], Kerner [8], and others, gave various derivations of this formula).

The convergence of the total-step iterative process (1) is quadratic. Using the Gauss–Seidel approach, (1) can be accelerated. In such a way, one obtains the single-step method

$$\hat{x}_k = x_k - \frac{\mathrm{P}(x_k)}{\prod_{i=1}^{k-1} (x_k - \hat{x}_i) \prod_{i=k+1}^{n} (x_k - x_i)} \quad (k = 1, \ldots, n). \tag{2}$$

It was proved in [2] that the R-order of convergence of procedure (2) is at least $r(n) = 1 + \sigma_n$, where $\sigma_n > 1$ is the unique positive solution of the equation $\sigma^n - \sigma - 1 = 0$.

A modification of SIP (1), which has cubic convergence and uses Weierstrass's correction W_i, is given by Nourein [12]:

$$\hat{x}_k = x_k - \frac{\mathrm{P}(x_k)}{\prod_{i=1,i\neq k}^{n} (x_k - x_i + W_i)} \quad (k = 1, \ldots, n). \tag{3}$$

Further acceleration of convergence can be attained combining the formulas (2) and (3) (see Petković and Milovanović [16]):

$$\hat{x}_k = x_k - \frac{\mathrm{P}(x_k)}{\prod_{i=1}^{k-1} (x_k - \hat{x}_i) \prod_{i=k+1}^{n} (x_k - x_i + W_i)} \quad (k = 1, \ldots, n). \tag{4}$$

The R-order of convergence of this iterative process is at least $r(n) = 1 + \tau_n$, where $\tau_n > 2$ is the unique positive solution of the equation

$$\tau^n - \tau - \textstyle\sum_{k=0}^{n-1} \tau^k = 0.$$

The modification of the basic Weierstrass formula (1) due to Börsch-Supan [4] follows:

$$\hat{x}_k = x_k - \frac{W_k}{1 + \sum\limits_{i=1,i\neq k}^{n} \dfrac{W_i}{x_k - x_i}} \quad (k = 1, \ldots, n). \tag{5}$$

The convergence order of this method is three.

Similar to constructing formula (3), the following method of the fourth order can be obtained from (5) (Nourein [13]):

$$\hat{x}_k = x_k - \frac{W_k}{1 + \sum\limits_{i=1,i\neq k}^{n} \dfrac{W_i}{x_k - W_k - x_i}} \quad (k = 1, \ldots, n). \tag{6}$$

3.2 Iterative Formulas with Derivatives

Using the logarithmic derivative of the polynomial P, Maehly [10] (and, later, Aberth [1], Börsch-Supan [3], Ehrlich [7], and others) derived the formula

$$\hat{x}_k = x_k - \frac{1}{N_k^{-1} - \sum_{i=1, i\neq k}^{n} \frac{1}{x_k - x_i}} \quad (k = 1, \ldots, n), \tag{7}$$

which has a cubic convergence.

The single-step modification of (7)

$$\hat{x}_k = x_k - \frac{1}{N_k^{-1} - \sum_{i=1}^{k-1} \frac{1}{x_k - \hat{x}_i} - \sum_{i=k+1}^{n} \frac{1}{x_k - x_i}} \quad (k = 1, \ldots, n) \tag{8}$$

has the R-order of convergence of at least $r(n) = 2 + \mu_n$, where $\mu_n > 1$ is the unique positive solution of the equation $\mu^n - \mu - 2 = 0$ (Alefeld and Herzberger [2]).

Using Newton's correction, Nourein [12] obtained the following modification of (7):

$$\hat{x}_k = x_k - \frac{1}{N_k^{-1} - \sum_{i=1, i\neq k}^{n} \frac{1}{x_k - x_i + N_i}} \quad (k = 1, \ldots, n), \tag{9}$$

with the convergence order equal to four.

Analogously to (4), we obtain from (8) the single-step method

$$\hat{x}_k = x_k - \frac{1}{N_k^{-1} - \sum_{i=1}^{k-1} \frac{1}{x_k - \hat{x}_i} - \sum_{i=k+1}^{n} \frac{1}{x_k - x_i + N_i}} \quad (k = 1, \ldots, n), \tag{10}$$

where the R-order of convergence is at least $r(n) = 2(1 + \lambda_n)$ and where $\lambda_n > 1$ is the unique positive solution of the equation $\lambda^n - \lambda - 1 = 0$ (Milovanović and Petković [11]).

The review of the number of basic arithmetic operations A(n), S(n), M(n), and D(n) for the aforementioned methods (1)–(10) (excluding only polynomial evaluations G(n)) is given in Table I as a function of polynomial degree n.

The values of the R-order of convergence $r(n)$ for the listed methods (1)–(10) are displayed in Table II for $n = 3(1)10$. The value of $r(n)$ when $n \to \infty$ is denoted by r_∞ (the last row).

4. EFFICIENCY OF SIMULTANEOUS METHODS

In order to compare the methods (1)–(10), we have computed the estimations of the efficiency for these methods using the definition (E_1). We have used the characteristics (period clocks) of the arithmetic units of the computing machines HONEYWELL DPS 6/92, VAX 11/780, IBM 4341, and, also, of the supercomputers CRAY X-MP/2 and FUJITSU VP-200 (on the basis of data given in [9]). Typical values of the operation weights, obtained after a procedure of normalization in relation to w_A, are given in Table III.

Table I. Number of Basic Arithmetic Operations for the Simultaneous Iterative Methods (1)–(10) as a Function of Polynomial Degree n

SIP	$A(n)$	$S(n)$	$M(n)$	$D(n)$
(1)	0	n^2	$n(n-2)$	n
(2)	0	n^2	$n(n-2)$	n
(3)	$n(n-1)$	$n(2n-1)$	$2n(n-2)$	$2n$
(4)	$\frac{1}{2}n(n-1)$	$2n^2-2n+1$	$2n^2-5n+2$	$2n-1$
(5)	$n(n-1)$	$2n^2-n$	$n(n-2)$	n^2+n
(6)	$n(n-1)$	$3n^2-2n$	$n(n-2)$	n^2+n
(7)	$n(n-1)$	$2n^2-n$	$n(n-1)$	n^2+n
(8)	$n(n-1)$	$2n^2-n$	$n(n-1)$	n^2+n
(9)	$2n(n-1)$	$2n^2-n$	$n(n-1)$	n^2+2n
(10)	$\frac{3}{2}n(n-1)$	$2n^2-n$	$n(n-1)$	n^2+2n-1

Table II. The Values of the R-Order of Convergence $r(n)$

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
$r(3)$	2	2.325	3	3.148	3	4	3	3.521	4	4.649
$r(4)$	2	2.221	3	3.066	3	4	3	3.353	4	4.441
$r(5)$	2	2.167	3	3.032	3	4	3	3.267	4	4.335
$r(6)$	2	2.135	3	3.016	3	4	3	3.215	4	4.269
$r(7)$	2	2.113	3	3.008	3	4	3	3.180	4	4.226
$r(8)$	2	2.097	3	3.004	3	4	3	3.154	4	4.194
$r(9)$	2	2.085	3	3.002	3	4	3	3.135	4	4.170
$r(10)$	2	2.076	3	3.001	3	4	3	3.125	4	4.152
r_∞	2	2.000	3	3.000	3	4	3	3.000	4	4.000

Table III. The Operation Weights Normalized in Relation to w_A

	w_A	w_S	w_M	w_D
HONEYWELL DPS 6/92	1.00	1.00	3.00	5.62
VAX 11/780	1.00	1.00	1.50	5.25
IBM 4341	1.00	1.00	1.50	12.37
CRAY X-MP/2	1.00	1.00	1.17	2.33
FUJITSU VP-200	1.00	1.00	1.33	9.33

The normalized cost of the evaluation $\theta(n)$ for all listed methods can be expressed in the form

$$\theta(n) = a + \frac{b}{n} + \frac{c}{n^2},$$

where a (>0), b, and c are real constants that depend on the number of basic arithmetic operations $A(n)$, $S(n)$, $M(n)$, $D(n)$ (Table I), $G(n) = (w_A + w_M)n^2$, and the operation weights w_A, w_S, w_M, and w_D (Table III). The values of the constants a, b, and c for the methods (1)–(10), related to the considered computers, are shown in Table IV.

Using the values of the R-order of convergence $r(n)$ (Table II) and the constants a, b, and c (Table IV), the coefficient of the efficiency $E(\text{SIP}, n)$ was computed by (E_1) for $n = 3(1)10$ and displayed in Table V for various

Table IV. The Values of the Constants a, b, and c

SIP	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
HONEYWELL DPS 6/92										
a	2.000	2.000	3.250	3.125	3.905	4.155	3.905	3.905	4.155	4.030
b	−0.095	−0.095	−0.069	−1.565	−0.595	−0.845	0.155	0.155	1.310	1.435
c	0.000	0.000	0.000	0.345	0.000	0.000	0.000	0.000	0.000	−1.405
VAX 11/780										
a	2.000	2.000	3.400	3.200	4.900	5.300	4.900	4.900	5.300	5.100
b	0.900	0.900	1.000	0.200	0.100	−0.300	0.700	0.700	2.400	2.600
c	0.000	0.000	0.000	−0.500	0.000	0.000	0.000	0.000	0.000	−2.100
IBM 4341										
a	2.000	2.000	3.400	3.200	7.748	8.148	7.748	7.748	8.148	7.948
b	3.748	3.748	6.696	5.896	2.948	2.548	3.548	3.548	8.096	8.296
c	0.000	0.000	0.000	−3.348	0.000	0.000	0.000	0.000	0.000	−4.948
CRAY X-MP/2										
a	2.000	2.000	3.461	3.230	3.995	4.456	3.995	3.995	4.456	4.226
b	−0.005	−0.005	−0.931	−1.701	−0.926	−1.387	−0.387	−0.387	0.226	0.456
c	0.000	0.000	0.000	0.465	0.000	0.000	0.000	0.000	0.000	−1.074
FUJITSU VP-200										
a	2.000	2.000	3.429	3.215	6.863	7.292	6.863	6.863	7.292	7.077
b	2.863	2.863	4.867	4.081	2.004	1.575	2.575	2.575	6.150	6.365
c	0.000	0.000	0.000	−2.434	0.000	0.000	0.000	0.000	0.000	−4.004

machines and SIP (1)–(10). The maximum values of $E(\text{SIP}, n)$ for each n are boxed. The arrows in the next to last column show the tendency of an increase ($\uparrow$) or a decrease ($\downarrow$) of $E(\text{SIP}, n)$ when n increases. If $n \to \infty$, then $\theta(n) \to a$ and $r(n) \to r_\infty = r$ ($r \in N$), where r is the convergence order of the basic method (in parallel fashion). Therefore, $E(\text{SIP}, n) \to r/a$ when $n \to \infty$ (the last column in Table V).

First of all, we observe from Table V that the order of the efficiency of SIP for the considered computers, stated according to the values $E(\text{SIP}, n)$, is preserving (with slight exceptions) varying n. This fact makes it possible for us to form a rating of the methods (1)–(10) related to their efficiency. Let $(E((k_1), n), \ldots, E((k_{10}), n))$ ($k_j \in \{1, \ldots, 10\}, j = 1, \ldots, 10$) be the ordered 10-tuplet whose components satisfy $E((k_1), n) > E((k_2), n) > \cdots > E((k_{10}), n)$, and let $R_{(E_1)} = (k_1, \ldots, k_{10})^T$ be the *rating vector* of the iterative methods $(k_1), \ldots, (k_{10})$ in reference to the definition (E_1). The vector $R_{(E_1)}$ defines the efficiency of SIP for a given computer. The rating of the considered methods, related to the given computers, for (relatively) small n (say, $n \leqq 10$) and for large n (>10) is shown in Table VI.

Remark 1. We observe from Table VI that there exists slight disagreement in the rating of the simultaneous methods for $n \leqq 10$ and for large n. The only exception is the HONEYWELL computer, but in that case, the differences between coefficients of efficiency of the dominant methods (2), (1), (10), (9), (6), and (4) are insignificant—mostly about 4 percent.

Remark 2. For all considered computers and for any polynomial degree n, SIP (2) is the most efficient, of course, in the sense of definition (E_1). Further, we

Table V. The Values of $E(\mathrm{SIP}, n)$ Computed by (E_1) (rounded to the third decimal digit)

	n								
SIP	3	4	5	6	7	8	9	10	$n \to \infty$
HONEYWELL DPS 6/92									
(1)	1.016	1.012	1.010	1.008	1.007	1.006	1.005	1.005	↓ 1.000
(2)	1.181	1.124	1.094	1.076	1.064	1.055	1.048	1.043	↓ 1.000
(3)	0.993	0.975	0.964	0.957	0.952	0.948	0.945	0.943	↓ 0.923
(4)	1.192	1.113	1.073	1.049	1.034	1.024	1.016	1.010	↓ 0.960
(5)	0.809	0.799	0.792	0.788	0.785	0.783	0.781	0.780	↓ 0.768
(6)	1.033	1.014	1.004	0.996	0.992	0.988	0.985	0.983	↓ 0.963
(7)	0.758	0.761	0.762	0.763	0.764	0.765	0.765	0.765	↑ 0.768
(8)	0.890	0.850	0.830	0.818	0.810	0.804	0.799	0.797	↓ 0.768
(9)	0.871	0.892	0.906	0.915	0.921	0.926	0.930	0.933	↑ 0.963
(10)	1.068	1.033	1.017	1.009	1.005	1.002	0.999	0.998	↓ 0.993
VAX 11/780									
(1)	0.870	0.899	0.917	0.930	0.940	0.947	0.952	0.957	↑ 1.000
(2)	1.011	0.998	0.994	0.993	0.993	0.993	0.993	0.994	↑ 1.000
(3)	0.804	0.822	0.833	0.841	0.847	0.851	0.854	0.857	↑ 0.882
(4)	0.980	0.953	0.942	0.937	0.935	0.934	0.933	0.934	↑ 0.937
(5)	0.608	0.609	0.610	0.610	0.610	0.611	0.611	0.611	↑ 0.612
(6)	0.769	0.766	0.763	0.762	0.761	0.760	0.760	0.759	↓ 0.755
(7)	0.584	0.591	0.595	0.598	0.600	0.601	0.603	0.604	↑ 0.612
(8)	0.686	0.661	0.648	0.641	0.636	0.632	0.630	0.629	↓ 0.612
(9)	0.656	0.678	0.692	0.702	0.709	0.714	0.719	0.722	↑ 0.755
(10)	0.811	0.790	0.783	0.780	0.779	0.778	0.777	0.778	↑ 0.784
IBM 4341									
(1)	0.616	0.681	0.727	0.762	0.789	0.810	0.828	0.842	↑ 1.000
(2)	0.716	0.756	0.788	0.813	0.833	0.850	0.863	0.874	↑ 1.000
(3)	0.533	0.591	0.633	0.664	0.689	0.708	0.724	0.737	↑ 0.882
(4)	0.657	0.687	0.714	0.738	0.757	0.773	0.787	0.799	↑ 0.937
(5)	0.344	0.354	0.350	0.364	0.367	0.370	0.372	0.373	↑ 0.387
(6)	0.445	0.455	0.462	0.467	0.470	0.472	0.474	0.476	↑ 0.491
(7)	0.336	0.347	0.355	0.360	0.363	0.366	0.368	0.370	↑ 0.387
(8)	0.394	0.388	0.386	0.386	0.385	0.385	0.385	0.386	↑ 0.387
(9)	0.369	0.393	0.410	0.421	0.430	0.437	0.442	0.447	↑ 0.491
(10)	0.458	0.457	0.461	0.464	0.468	0.471	0.473	0.476	↑ 0.503
CRAY X-MP/2									
(1)	1.001	1.001	1.001	1.000	1.000	1.000	1.000	1.000	↓ 1.000
(2)	1.163	1.111	1.084	1.068	1.057	1.049	1.043	1.038	↓ 1.000
(3)	0.952	0.923	0.916	0.907	0.902	0.897	0.893	0.891	↓ 0.867
(4)	1.159	1.082	1.042	1.019	1.004	0.993	0.985	0.979	↓ 0.929
(5)	0.814	0.797	0.787	0.781	0.777	0.773	0.771	0.769	↓ 0.751
(6)	1.002	0.973	0.957	0.947	0.939	0.934	0.930	0.927	↓ 0.898
(7)	0.776	0.769	0.766	0.763	0.761	0.760	0.759	0.758	↓ 0.751
(8)	0.911	0.860	0.834	0.818	0.807	0.799	0.793	0.790	↓ 0.751
(9)	0.883	0.886	0.888	0.890	0.891	0.892	0.893	0.893	↑ 0.898
(10)	1.092	1.039	1.014	0.999	0.990	0.983	0.978	0.974	↓ 0.947
FUJITSU VP-200									
(1)	0.677	0.736	0.777	0.807	0.830	0.848	0.863	0.875	↑ 1.000
(2)	0.787	0.818	0.842	0.862	0.877	0.889	0.899	0.908	↑ 1.000
(3)	0.594	0.646	0.681	0.707	0.727	0.743	0.756	0.766	↑ 0.875
(4)	0.731	0.751	0.771	0.788	0.803	0.815	0.825	0.834	↑ 0.933
(5)	0.398	0.407	0.413	0.417	0.420	0.422	0.423	0.425	↑ 0.437
(6)	0.512	0.521	0.526	0.529	0.532	0.534	0.536	0.537	↑ 0.549
(7)	0.388	0.400	0.407	0.411	0.415	0.418	0.420	0.421	↑ 0.437
(8)	0.456	0.447	0.443	0.441	0.440	0.439	0.439	0.438	↓ 0.437
(9)	0.428	0.453	0.469	0.481	0.490	0.496	0.502	0.506	↑ 0.549
(10)	0.531	0.527	0.529	0.532	0.535	0.537	0.539	0.541	↑ 0.565

Table VI. The Rating Vectors $R_{(E_1)}$

HONEYWELL DPS 6/92		VAX 11/780		IBM 4341		CRAY X-MP/2		FUJITSU VP-200	
$n \leqq 10$	large n	$n \leqq 10$	large n	$n \leqq 10$	large n	$n \leqq 10$	large n	$n \leqq 10$	large n
2	2	2	2	2	2	2	2	2	2
4	1	4	1	1	1	4	1	1	1
10	10	1	4	4	4	1	10	4	4
1	9	3	3	3	3	10	4	3	3
6	6	10	10	6	10	6	6	10	10
3	4	6	6	10	6	3	9	6	6
9	3	9	9	9	9	9	3	9	9
8	8	8	8	8	8	8	8	8	8
5	5	5	5	5	5	5	5	5	5
7	7	7	7	7	7	7	7	7	7

observe that the class of methods of Weierstrass's type (SIP (1)–(4)) is the most powerful (in the order (2), (1), (4), and (3)). Besides, it is obvious from Table VI that the methods (7), (5), and (8) are the least efficient for all considered computers. These conclusions have been verified by the experimental results, which were based on the CPU times of computers.

In order to verify the previous results related to the coefficient of the efficiency of SIP (1)–(10), we have performed the analysis of the efficiency of SIP by measuring the CPU times corresponding to HONEYWELL DPS 6/92, VAX 11/780, and IBM 4341. In this experiment the iterative methods were tested on polynomials with degrees ranging from 4 to 15. The (real) zeros of those polynomials were normalized to lie in the union of intervals $[-5.0, -0.5] \cup [0.5, 5.0]$. The criterion for stopping any iterative process was given by

$$\text{(SC)} \qquad \max_{1 \leq i \leq n} |x_i^m - \xi_i| < \epsilon = 10^{-q},$$

where m is the iteration index and q is the number of correct decimal places in the approximations to the zeros $\xi_1, \ldots, \xi_n$. Two values of the required accuracy ϵ were used, $\epsilon_1 = 10^{-12}$ and $\epsilon_2 = 10^{-30}$. To avoid round-off errors and to attain very high accuracy of approximations, all programs were realized in multiple-precision arithmetic.

Let T_k be the CPU time necessary that SIP$(k)(k \in \{1, \ldots, 10\})$ satisfies the stopping criterion (SC). Further, let $T_{\text{CPU}} = (T_{k_1}, \ldots, T_{k_{10}})$ be the ordered 10-tuplet of CPU times such that $T_{\min} = T_{k_1} < T_{k_2} < \cdots < T_{k_{10}}$, and let $t_{\text{CPU}} = (t_{k_1}, \ldots, t_{k_{10}})$, where t_{k_j} is obtained by normalization, $t_{k_j} = T_{k_j}/T_{\min}$ $(j = 1, \ldots, 10)$ (consequently, $t_{k_1} = 1$). Now we can correspond to t_{CPU} the rating vector $R_{\text{CPU}} = (k_1, \ldots, k_{10})^T$, which determines the efficiency of SIP (1)–(10) related to the CPU times.

The tests performed on the aforementioned three computing machines have demonstrated very good coincidence of the rating vectors $R_{(E_1)}$ and R_{CPU}. This means that the definition (E_1) for the coefficient of the efficiency of SIP is really applicable and describes a real situation in practical realization of an iterative process on a computer.

Table VII. The Normalized CPU Times

SIP	Number of iterations required for accuracy ϵ_1	ϵ_2	(SC): $\epsilon_1 = 10^{-12}$ HONEYWELL DPS 6/92	VAX 11/780	IBM 4341	(SC): $\epsilon_2 = 10^{-30}$ HONEYWELL DPS 6/92	VAX 11/780	IBM 4341
(1)	4	6	1.0022	1.0031	1.0025	1.2027	1.2037	2.2030
(2)	4	5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
(3)	3	4	1.2143	1.4535	1.4619	1.2952	1.5504	1.5593
(4)	3	4	1.0474	1.3570	1.2014	1.1173	1.4474	1.2815
(5)	3	4	1.4408	1.7674	2.1759	1.5369	1.8853	2.3210
(6)	2	3	1.0054	1.2403	1.5014	1.2065	1.4884	1.8018
(7)	3	4	1.4089	1.5837	2.1111	1.5028	1.6893	2.2519
(8)	3	4	1.4072	1.5814	2.1092	1.5010	1.6868	2.2498
(9)	2	3	1.0459	1.1977	1.6140	1.2551	1.4372	1.9369
(10)	2	3	1.0109	1.1504	1.5572	1.2131	1.3805	1.8687

Table VIII. The Rating Vectors R_{CPU}

(SC): $\epsilon_1 = 10^{-12}$ HONEYWELL DPS 6/92	VAX 11/780	IBM 4341	(SC): $\epsilon_2 = 10^{-30}$ HONEYWELL DPS 6/92	VAX 11/780	IBM 4341
2	2	2	2	2	2
1	1	1	4	1	1
6	10	4	1	10	4
10	9	3	6	9	3
9	6	6	10	4	6
4	4	10	9	6	10
3	3	9	3	3	9
8	8	8	8	8	8
7	7	7	7	7	7
5	5	5	5	5	5

To demonstrate the above conclusions, we give the normalized CPU times t_{k_j} for the computers HONEYWELL DPS 6/92, VAX 11/780, and IBM 4341. These times have been calculated by normalizing the measured CPU times T_{k_j} ($j = 1, \ldots, 10$) applying the iterative methods (1)–(10) for the determination of zeros of the polynomial

$$\begin{aligned} P(x) &= x^5 - 15x^4 + 85x^3 - 225x^2 + 274x - 120 \\ &= (x-1)(x-2)(x-3)(x-4)(x-5). \end{aligned}$$

The initial approximations were correct to one decimal place. The normalized CPU times for two SCs, $\epsilon_1 = 10^{-12}$ and $\epsilon_2 = 10^{-30}$, are displayed in Table VII.

According to the values of the normalized CPU times, shown in Table VII, the rating vectors R_{CPU} for the tested computers were formed relating to two SCs and given in Table VIII. Comparing the rating vectors R_{CPU} and $R_{(E_1)}$ (Table VI), we notice significant coincidence of the corresponding components of these vectors. For example, eight components of R_{CPU} and $R_{(E_1)}$ coincide for the IBM 4341 system. This conclusion refers to the above example where $n = 5$, but the same is valid for other n too.

In conclusion, the definition E_1 for the efficiency index of SIP is practically applicable. In particular, this definition and the corresponding rating vectors $R_{(E_1)}$ can be of interest in designing a package for the simultaneous approximation of polynomial zeros, in which automatic procedure selection is desired.

ACKNOWLEDGMENT

The authors are grateful to the referee for helpful suggestions and valuable comments.

REFERENCES

1. ABERTH, O. Iteration methods for finding all zeros of a polynomial simultaneously. *Math. Comput. 27* (1973), 339–344.
2. ALEFELD, G., AND HERZBERGER, J. On the convergence speed of some algorithms for the simultaneous approximation of polynomial roots. *SIAM J. Numer. Anal. 11*, 2 (1974), 237–243.
3. BÖRSCH-SUPAN, W. A posteriori error bounds for the zeros of polynomials. *Numer. Math. 5*, 4 (1963), 380–398.
4. BÖRSCH-SUPAN, W. Residuenabschätzung für Polynom-Nullstellen mittels Lagrange-Interpolation. *Numer. Math. 14*, 3 (1970), 287–296.
5. DOČEV, K. Vidoizmenen metod na Newton za edinovremenno priblizitel'no presmyatane na vsichki koreni na dadeno algebrichno uravnenie. *Fiz.-Mat. Spis. Bulgar. Akad. Nauk 5*, 2 (1962), 136–139.
6. DURAND, E. *Solution Numérique des Équations Algébraique* (tome 1). Masson et Compagnie, Paris, 1960.
7. EHRLICH, L. W. A modified Newton method for polynomials. *Commun. ACM 10*, 2 (Feb. 1967), 107–108.
8. KERNER, I. O. Ein Gesamtschrittverfahren zur Berechnung der Nullstellen von Polynomen. *Numer. Math. 8*, 3 (1966), 290–294.
9. LUBECK, O., MOORE, J., AND MENDEZ, R. A benchmark comparison of three supercomputers: Fujitsu VP-200, Hitachi S810/20, and Cray X-MP/2. *Computer 18*, 12 (1985), 10–24.
10. MAEHLY, H. J. Zur iterativen Auflösung algebraischer Gleichungen. *Z. Angew. Math. Phys. 5* (1954), 260–263.
11. MILOVANOVIĆ, G. V., AND PETKOVIĆ, M. S. On the convergence order of a modified method for simultaneous finding polynomial zeros. *Computing 30*, 2 (1983), 171–178.
12. NOUREIN, A. W. M. An improvement on two iteration methods for simultaneous determination of the zeros of a polynomial. *Int. J. Comput. Math. 6*, 3 (1977), 241–252.
13. NOUREIN, A. W. M. An improvement on Nourein's method for the simultaneous determination of the zeroes of a polynomial (an algorithm). *J. Comput. Appl. Math. 3*, 2 (1977), 109–110.
14. ORTEGA, J. M., AND RHEINBOLDT, W. C. *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York, 1970.
15. OSTROWSKI, A. *Solution of Equations and Systems of Equations.* Academic Press, New York, 1966.
16. PETKOVIĆ, M. S., AND MILOVANOVIĆ, G. V. A note on some improvements of the simultaneous methods for determination of polynomial zeros. *J. Comput. Appl. Math. 9*, 1 (1983), 65–69.
17. TRAUB, J. F. *Iterative Methods for the Solution of Equations.* Prentice-Hall, Englewood Cliffs, N. J., 1964.
18. WEIERSTRASS, K. Neuer Beweis des Satzes, dass jede Ganze Rationale Function einer Veränderlichen dargestellt werden kann als ein Product aus Linearen Functionen darselben Veränderlichen. *Ges. Werke 3* (1903), 251–269.

Received July 1985; revised September 1986; accepted September 1986