Journal of Computational Mathematics, Vol.8, No.3, 1990, 202–211.

COMPARATIVE TESTING OF FIVE NUMERICAL METHODS FOR FINDING ROOTS OF POLYNOMIALS*

Glenn R. Luecke
(*Department of Mathematics, Computation Center, Iowa State University, U.S.A.*)
James D. Francis
(*Department of Mathematics, University of Washington, U.S.A.*)

Abstract

This paper summarizes the results of the comparative testing of (1) Wilf's global bisection method, (2) the Laguerre method, (3) the companion matrix eigenvalue method, (4) the companion matrix eigenvalue method with balancing, and (5) the Jenkens-Traub method, all of which are methods for finding the zeros of polynomials. The test set of polynomials used are those suggested by [5]. The methods were compared on each test polynomial on the basis of the accuracy of the computed roots and the CPU time required to numerically compute all roots.

Introduction

This paper summarizes the results of comparative testing of five methods that find all zeros of a polynomial. Twenty-five polynomials were used which were designed to test potential weakness in such algorithms; see [5]. All computer runs were made on a National Advanced Systems AS/6 computer using the SVS operating system and using the WATFIV FORTRAN compiler in double precision, which means about fifteen decimal digits of accuracy.

Methods tested

1. The Jenkins–Traub Method (JTM)

A description of this method can be found in [3]. The IMSL Library [2] routine ZRPOLY was used to implement this method for polynomials with real coefficients, and the IMSL Library routine ZCPOLY was used for polynomials with complex coefficients.

2. The Laguerre Method (LM)

A description of this method can be found in [1]. The IMSL Library routine ZPOLR is an adaptation of the program ZERPOL developed by B.T.Smith [7]. This routine will only find approximations to the roots of polynomials with real coefficients.

3. The Eigenvalue Method (EM)

It is well known that if A is the $n \times n$ companion matrix of a polynomial p of degree n, then the characteristic polynomial os A is a known scalar multiple of p; see [6]. Thus, the eigenvalues of A are the roots of p. Since A is an Upper Hessenberg matrix, the IMSL Library routines EQRH3F (for real A) and ELRHIC (for complex A) were used to compute the eigenvalues of A.

4. The Eigenvalue Method with Balancing (EMB)

This method is identical to the companion matrix eigenvalue method mentioned above, except that the matrix A was balanced (see [2]) before the eigenvalues of A were computed.

* Received October 4, 1987.

The IMSL Library routines EBALAF (for real A) and EBALAC (for complex A) were used to balance A.

5. The Wilf Method (WM)

A description of this method can be found in [8]. The computer algorithm for this method was kindly sent to us by Dr. Herbert Wilf. In the process of testing this routine, several bugs were found in the FORTRAN code received and appropriate corrections were made.

Test Polynomials

Test polynomials used are those suggested by [5]. Each polynomial was designed to test for a specific potential problem. Even though this report gives some of the test polynomials in factored form, the factors were multiplied out exactly and the coefficients of the polynomial in the form $a_0 + a_1x + a_2x^2 + \cdots a_nx^n$ were used by each method to determine the roots. The following is a list of fourteen polynomials $p_1(z), \cdots, p_{14}(z)$ from which the twenty-five test polynomials $Ex1, \cdots, Ex25$ are derived.

(1) $p_1(z) = B(z-A)(z+A)(z-1)$ with $A = 10^{-2}$, $B = 1$ (Ex 1); $A = 10^{-6}$, $B = 1$ (Ex 2); $A = 10^8$, $B = 1$ (Ex 3); $A = 2, B = 10^{-6}$ (Ex 4); $A = 2, B = 10^8$ (Ex 5). $p_1(z)$ is designed to test whether relatively large or small zeros or whether large or small coefficients cause difficulty for the method; see [5].

(2) $p_2(z) = (z-1)(z-2)(z-3)\cdots(z-N)$ with $N = 5$ (Ex 6) and $N = 10$ (Ex7). This polynomial is ill-conditioned for large values of N in the sense that the magnitudes of the coefficients vary considerably. The leading coefficient is one while the constant term is $N!$. This causes extreme variation of the polynomial between consecutive roots which can affect the convergence of some methods. N should be chosen small enough so that all coefficients can be represented exactly.

(3) $p_3(z) = (z-10^{-1})(z-10^{-2})\cdots(z-10^{-N})$ with $N = 3$ (Ex 8) and 4 (Ex 9). For this polynomial the coefficient of z^n is one and the constant coefficient is $10^{-N!}$.$p_3(z)$ is designed to test for underflow in its evaluation and the ability of a method to distinguish zeros that are close together.

(4) $p_4(z) = (z-.1)^3(z-.5)(z-.6)(z-.7)$, (Ex 10). This polynomial, along with p_5 through p_8, have one or more multiple roots and/or "nearly" multiple roots (i.e. distinct, but nearly equal roots), which can cause convergence difficulties for many algorithms.

(5) $p_5(z) = (z-.1)^4(z-.2)^3(z-.3)^2(z-.4)$, (Ex 11).

(6) $p_6(z) = (z-.1)(z-1.001)(z-.998)(z-1.00002)(z-.99999)$, (Ex 12).

(7) $p_7(z) = (z-.001)(z-.01)(z-.1)(z-1)(z-10)(z-(.1+Ai))(z-(.1-Ai))$ with $A = 0$, (Ex13), $A = .1$ (Ex14), $A = .001$ (Ex15). This polynomial has a multiple root at 0.1 when $A = 0$ and a "nearly" multiple root when A is small.

(8) $p_8(z) = (z+1)^5$, (Ex 16).

(9) $p_9(z) = z^5 - 1$, (Ex 17). The five roots of this polynomial are equimodular roots which can cause convergence difficulties, especially for algorithms which use power techniques to separate such roots.

(10) $p_{10}(z) = (z^5 - A^{-1})(z^5 + A)$ with $A = 10^5$ (Ex 18) and 10^{10} (Ex19).

(11) $p_{11}(z) = (z-A)(z-1)(z-A^{-1})$ with $A = 10^3$ (Ex20) and 10^6 (Ex21). p_{11} is designed to test the accuracy of methods that compute roots one (or one complex pair) at a time and then use deflation to calculate the rest.

(12) $p_{12}(z) = \prod_{k=1-m}^{m-1} (z - \exp(ik\pi/2m)) \prod_{k=m}^{3m} (z - .9\exp(ik\pi/2m))$ with $m = 1$ (Ex22) and 2 (Ex23). p_{12} is designed to test for deflation accuracy (as was p_{11}).

(13) $p_{13}(z) = (z - (1+i))^2(z - (4+3i))(z - (4-3i))(z - (3.999+3i))$, (Ex 24).

(14) $p_{14}(z) = (z - (1+i))^2(z - (-1-2i))(z - (1-i))$, (Ex 25). $p_{13}(z)$ and $p_{14}(z)$ have complex roots and complex coefficients and were included to have some comparisons of the methods for polynomials of this type. The IMSL routine for Laguerre's method would not accept polynomials with complex coefficients.

Results of Testing

Each method produced the correct number of roots except Wilf's method. Wilf's method probably would have produced the correct number of roots for each test polynomial if the job had not been cancelled due to excessive execution time. Table 1 gives the number of roots produced by Wilf's method (W-Roots), the correct number of roots (Roots), the execution time in seconds for Wilf's method (W-CPU), and the largest execution time (CPU) in seconds of all methods excluding Wilf's method. Only the examples for which Wilf's method did not produce all roots are listed.

Table 1. Examples for which Wilf's Method did not produce all roots

	W-Roots	Roots-CPU	CPU	
Ex1	0	3	80	.035
Ex2	0	3	80	.035
Ex4	1	3	80	.035
Ex9	2	3	30	.056
Ex12	4	5	30	.235
Ex20	1	10	90	1.26
Ex21	1	10	90	1.24
Ex22	2	3	60	.029
Ex23	0	3	60	.021

The exact roots are known for each test polynomials so the absolute and relative error were calculated for each root. The following tables report the maximum absolute error, the minimum absolute error, the average absolute error, and the average relative error for each method applied to each example. "∞" in the chart means that the method did not produce enough roots to calculate the error. "NA" for some of the Laguerre's method results means "Not Appropriate" since the IMSL version of Laguerre's method could not handle polynomials with complex coefficients. Recall

"JTM" stands for the Jenkins-Traub Method,

"EM" Stands for the Eigenvalue Method,

"EMB" stands for te Eigenvalue Method with Balancing,

"LM" stands for the Laguerre Method, and

"WM" Stands for the Wilf Method.

The table titles should be self-explanatory. For example, "maximum absolute error" means each number reported is the maximum of the absolute errors calculated for each root of the given test polynomial. All numbers reported are rounded.

Table 2. Maximum Absolute Error

	JTM	EM	EMB	LM	WM
Ex1	1.2×10^{-26}	9.9×10^{-9}	1.4×10^{-15}	2.0×10^{-16}	∞
Ex2	9.0×10^{-21}	6.7×10^{-11}	1.1×10^{-15}	2.0×10^{-16}	∞
Ex3	0.0×10^{-0}	1.0×10^{-0}	1.4×10^{-7}	0.0×10^{-0}	6.5×10^{-8}
Ex4	0.0×10^{-0}	8.0×10^{-15}	8.0×10^{-15}	0.0×10^{-0}	∞
Ex5	0.0×10^{-0}	7.0×10^{-15}	7.0×10^{-15}	0.0×10^{-0}	3.8×10^{-17}
Ex6	1.7×10^{-14}	2.4×10^{-13}	1.0×10^{-12}	1.2×10^{-13}	0.0×10^{-0}
Ex7	3.4×10^{-11}	1.9×10^{-9}	1.5×10^{-8}	6.0×10^{-10}	5.8×10^{-10}
Ex8	1.0×10^{-17}	9.2×10^{-15}	3.0×10^{-16}	1.0×10^{-17}	0.0×10^{-0}
Ex9	1.0×10^{-17}	1.2×10^{-7}	1.2×10^{-7}	1.0×10^{-17}	∞
Ex10	3.6×10^{-8}	6.7×10^{-6}	2.0×10^{-6}	7.8×10^{-7}	1.2×10^{-6}
Ex11	6.5×10^{-5}	1.4×10^{-3}	4.3×10^{-4}	4.4×10^{-4}	6.0×10^{-5}
Ex12	7.7×10^{-4}	4.0×10^{-5}	4.0×10^{-5}	4.0×10^{-4}	∞
Ex13	1.4×10^{-7}	2.5×10^{-5}	8.9×10^{-7}	1.7×10^{-6}	7.4×10^{-7}
Ex14	7.1×10^{-17}	7.6×10^{-12}	7.4×10^{-14}	3.1×10^{-15}	0.0×10^{-0}
Ex15	2.4×10^{-13}	1.6×10^{-11}	1.6×10^{-11}	6.0×10^{-13}	3.2×10^{-13}
Ex16	6.0×10^{-8}	1.4×10^{-3}	7.8×10^{-4}	2.8×10^{-3}	1.0×10^{-3}
Ex17	5.0×10^{-16}	2.6×10^{-11}	2.6×10^{-11}	2.9×10^{-16}	3.7×10^{-3}
Ex18	2.3×10^{-15}	2.6×10^{-10}	2.6×10^{-10}	3.1×10^{-15}	∞
Ex19	3.6×10^{-14}	2.3×10^{-1}	1.4×10^{-15}	3.0×10^{-14}	∞
Ex20	2.0×10^{-13}	1.1×10^{-12}	1.1×10^{-12}	2.0×10^{-19}	∞
Ex21	3.0×10^{-22}	2.0×10^{-10}	1.3×10^{-9}	0.0×10^{-0}	∞
Ex22	2.0×10^{-16}	5.4×10^{-15}	5.4×10^{-15}	6.0×10^{-16}	3.9×10^{-15}
Ex23	1.5×10^{-13}	1.6×10^{-13}	1.6×10^{-13}	1.5×10^{-13}	2.2×10^{-13}
Ex24	9.4×10^{-10}	5.0×10^{-8}	7.1×10^{-8}	NA	4.6×10^{-8}
Ex25	8.8×10^{-10}	1.3×10^{-8}	8.0×10^{-9}	NA	1.2×10^{-8}

Table 3. Minimum Absolute Error

	JTM	EM	EMB	LM	WM
Ex1	0.0×10^{-0}	5.0×10^{-16}	8.0×10^{-28}	2.0×10^{-28}	∞
Ex2	0.0×10^{-0}	2.2×10^{-15}	1.0×10^{-21}	0.0×10^{-0}	∞
Ex3	0.0×10^{-0}	5.0×10^{-1}	0.0×10^{-0}	0.0×10^{-0}	0.0×10^{-0}
Ex4	0.0×10^{-0}	0.0×10^{-0}	0.0×10^{-0}	0.0×10^{-0}	0.0×10^{-0}
Ex5	0.0×10^{-0}	0.0×10^{-0}	0.0×19^{-0}	0.0×10^{-0}	1.7×10^{-27}
Ex6	1.0×10^{-15}	4.2×10^{-15}	9.0×10^{-15}	0.0×10^{-0}	0.0×10^{-0}
Ex7	4.0×10^{-15}	2.3×10^{-14}	4.0×10^{-15}	9.0×10^{-15}	0.0×10^{-0}
Ex8	0.0×10^{-0}	1.6×10^{-15}	3.0×10^{-19}	0.0×10^{-0}	0.0×10^{-0}
Ex9	0.0×10^{-0}	1.1×10^{-8}	1.1×10^{-8}	0.0×10^{-0}	∞
Ex10	3.9×10^{-13}	2.4×10^{-13}	4.8×10^{-14}	2.2×10^{-13}	0.0×10^{-0}
Ex11	3.0×10^{-9}	5.3×10^{-10}	5.0×10^{-11}	7.8×10^{-11}	4.1×10^{-13}
Ex12	3.4×10^{-14}	1.3×10^{-15}	1.3×10^{-15}	3.4×10^{-14}	∞
Ex13	0.0×10^{-0}	1.1×10^{-14}	8.0×10^{-18}	0.0×10^{-0}	3.6×10^{-24}
Ex14	0.0×10^{-0}	1.1×10^{-14}	5.9×10^{-18}	0.0×10^{-0}	0.0×10^{-0}
Ex15	0.0×10^{-0}	1.0×10^{-14}	0.0×10^{-0}	0.0×10^{-0}	0.0×10^{-0}
Ex16	0.0×10^{-0}	1.4×10^{-3}	7.7×10^{-4}	1.7×10^{-3}	2.9×10^{-4}
Ex17	9.7×10^{-17}	1.4×10^{-15}	1.4×10^{-15}	0.0×10^{-0}	5.4×10^{-14}
Ex18	0.0×10^{-0}	0.0×10^{-0}	0.0×10^{-0}	1.6×10^{-16}	∞
Ex19	0.0×10^{-0}	1.0×10^{-2}	0.0×10^{-0}	0.0×10^{-0}	∞
Ex20	0.0×10^{-0}	2.0×10^{-19}	2.0×10^{-19}	0.0×10^{-0}	∞
Ex21	0.0×10^{-0}	3.1×10^{-20}	3.0×10^{-22}	0.0×10^{-0}	∞
Ex22	0.0×10^{-0}	2.6×10^{-15}	2.6×10^{-15}	3.0×10^{-16}	0.0×10^{-0}
Ex23	0.0×10^{-0}	1.6×10^{-15}	1.6×10^{-15}	0.0×10^{-0}	0.0×10^{-0}
Ex24	1.4×10^{-15}	9.2×10^{-15}	6.0×10^{-15}	NA	0.0×10^{-0}
Ex25	2.0×10^{-16}	2.9×10^{-15}	3.2×10^{-15}	NA	0.0×10^{-0}

Table 4. Average Absolute Error

	JTM	EM	EMB	LM	WM
Ex1	4.3×10^{-27}	6.6×10^{-9}	4.7×10^{-16}	6.7×10^{-17}	∞
Ex2	4.3×10^{-23}	4.4×10^{-11}	1.5×10^{-15}	6.7×10^{-17}	∞
Ex3	0.0×10^{-0}	6.7×10^{-1}	9.0×10^{-8}	0.0×10^{-0}	2.2×10^{-8}
Ex4	0.0×10^{-0}	5.3×10^{-15}	5.3×10^{-15}	0.0×10^{-0}	∞
Ex5	0.0×10^{-0}	4.7×10^{-15}	4.7×10^{-15}	0.0×10^{-0}	1.8×10^{-18}
Ex6	7.0×10^{-15}	1.3×10^{-13}	4.4×10^{-13}	5.3×10^{-14}	0.0×10^{-0}
Ex7	1.8×10^{-11}	6.4×10^{-10}	5.0×10^{-9}	1.9×10^{-9}	1.6×10^{-10}
Ex8	3.3×10^{-18}	6.0×10^{-15}	1.1×10^{-6}	4.1×10^{-18}	0.0×10^{-0}
Ex9	2.6×10^{-18}	6.8×10^{-8}	6.8×10^{-8}	3.1×10^{-18}	∞
Ex10	1.8×10^{-8}	3.4×10^{-6}	1.0×10^{-6}	3.9×10^{-7}	4.9×10^{-7}
Ex11	3.8×10^{-5}	8.1×10^{-4}	2.4×10^{-4}	2.2×10^{-4}	2.6×10^{-5}
Ex12	3.1×10^{-4}	1.6×10^{-5}	1.6×10^{-5}	1.7×10^{-4}	∞
Ex13	4.4×10^{-8}	1.1×10^{-5}	3.8×10^{-7}	5.8×10^{-7}	3.1×10^{-7}
Ex14	1.3×10^{-17}	2.4×10^{-12}	1.2×10^{-14}	5.2×10^{-16}	0.0×10^{-0}
Ex15	6.7×10^{-14}	4.5×10^{-9}	2.5×10^{-12}	1.7×10^{-13}	7.4×10^{-14}
Ex16	2.5×10^{-8}	1.4×10^{-3}	7.8×10^{-4}	2.3×10^{-3}	4.4×10^{-4}
Ex17	3.0×10^{-16}	2.1×10^{-11}	2.1×10^{-11}	2.1×10^{-16}	7.3×10^{-4}
Ex18	1.1×10^{-15}	1.0×10^{-10}	1.0×10^{-10}	9.5×10^{-16}	∞
Ex19	1.1×10^{-14}	1.2×10^{-1}	2.7×10^{-6}	8.3×10^{-15}	∞
Ex20	6.7×10^{-14}	3.7×10^{-13}	3.7×10^{-13}	6.7×10^{-20}	∞
Ex21	1.0×10^{-22}	6.7×10^{-11}	4.3×10^{-10}	0.0×10^{-0}	∞
Ex22	1.5×10^{-16}	3.9×10^{-15}	3.9×10^{-15}	4.5×10^{-16}	2.0×10^{-15}
Ex23	3.8×10^{-14}	4.5×10^{-15}	4.5×10^{-15}	3.8×10^{-14}	5.7×10^{-14}
Ex24	2.2×10^{-10}	1.0×10^{-8}	2.0×10^{-8}	NA	1.8×10^{-8}
Ex25	4.4×10^{-10}	6.5×10^{-9}	4.2×10^{-9}	NA	5.6×10^{-9}

Table 5. Average Relative Error

	JTM	EM	EMB	LM	WM
Ex1	4.3×10^{-15}	$6.7 \times 10^{+3}$	1.1×10^{-15}	2.0×10^{-16}	∞
Ex2	4.3×10^{-17}	4.4×10^{-5}	1.1×10^{-15}	6.7×10^{-17}	∞
Ex3	0.0×10^{-0}	3.3×10^{-1}	9.0×10^{-16}	0.0×10^{-17}	2.2×10^{-16}
Ex4	0.0×10^{-0}	2.7×10^{-15}	2.7×19^{-15}	0.0×10^{-0}	∞
Ex5	0.0×10^{-0}	2.3×10^{-15}	2.3×19^{-15}	0.0×10^{-0}	8.9×10^{-18}
Ex6	2.6×10^{-15}	3.8×10^{-14}	1.1×10^{-13}	1.5×10^{-14}	0.0×10^{-0}
Ex7	2.9×10^{-12}	9.8×10^{-11}	7.0×10^{-10}	2.8×10^{-10}	2.2×10^{-11}
Ex8	3.3×10^{-17}	2.7×10^{-12}	2.0×10^{-15}	1.0×10^{-16}	0.0×10^{-0}
Ex9	2.6×10^{-17}	6.2×10^{-5}	6.2×10^{-5}	7.5×10^{-17}	∞
Ex10	1.8×10^{-7}	3.4×10^{-5}	1.0×10^{-5}	3.9×10^{-6}	4.9×10^{-6}
Ex11	3.1×10^{-4}	6.8×10^{-3}	2.0×10^{-3}	1.5×10^{-3}	2.3×10^{-4}
Ex12	3.1×10^{-4}	1.6×10^{-5}	1.6×10^{-5}	1.7×10^{-4}	∞
Ex13	4.4×10^{-7}	1.1×10^{-4}	3.8×10^{-6}	5.8×10^{-6}	3.1×10^{-6}
Ex14	1.5×10^{-16}	9.6×10^{-10}	4.2×10^{-15}	7.5×10^{-16}	0.0×10^{-0}
Ex15	6.7×10^{-13}	4.5×10^{-8}	1.9×10^{-10}	1.7×10^{-12}	7.4×10^{-13}
Ex16	2.5×10^{-8}	1.4×10^{-3}	7.8×10^{-4}	2.3×10^{-3}	4.4×10^{-4}
Ex17	3.0×10^{-16}	2.1×10^{-11}	2.1×10^{-11}	2.1×10^{-16}	7.3×10^{-4}
Ex18	1.8×10^{-16}	2.1×10^{-11}	2.1×10^{-11}	4.7×10^{-16}	∞
Ex19	2.4×10^{-16}	5.1×10^{-1}	2.7×10^{-8}	1.0×10^{-15}	∞
Ex20	6.7×10^{-17}	1.3×10^{-15}	1.3×10^{-15}	6.7×10^{-17}	∞
Ex21	1.0×10^{-16}	3.4×10^{-13}	1.5×10^{-15}	0.0×10^{-0}	∞
Ex22	1.6×10^{-16}	4.1×10^{-15}	4.1×10^{-15}	4.8×10^{-16}	2.2×10^{-15}
Ex23	3.8×10^{-14}	4.6×10^{-14}	4.6×10^{-14}	3.8×10^{-14}	5.7×10^{-14}
Ex24	1.5×10^{-10}	1.4×10^{-8}	1.4×10^{-8}	NA	1.3×10^{-8}
Ex25	3.1×10^{-10}	4.6×10^{-9}	3.0×10^{-9}	NA	4.0×10^{-9}

Table 6. CPU Time in Seconds

	JTM	EM	EMB	LM	WM
Ex1	1.2×10^{-2}	1.0×10^{-2}	1.9×10^{-2}	3.5×10^{-2}	$8.0 \times 10^{+1}$
Ex2	1.2×10^{-2}	1.6×10^{-2}	2.4×10^{-2}	3.5×10^{-2}	$8.0 \times 10^{+1}$
Ex3	1.1×10^{-2}	7.0×10^{-2}	2.2×10^{-2}	1.8×10^{-2}	4.6×10^{-0}
Ex4	1.3×10^{-2}	2.9×10^{-2}	3.3×10^{-2}	3.5×10^{-2}	$8.0 \times 10^{+1}$
Ex5	1.4×10^{-2}	2.9×10^{-2}	3.3×10^{-2}	3.5×10^{-2}	5.2×10^{-0}
Ex6	3.9×10^{-2}	1.2×10^{-1}	1.4×10^{-1}	1.0×10^{-2}	7.9×10^{-0}
Ex7	1.3×10^{-1}	6.8×10^{-1}	8.1×10^{-1}	2.8×10^{-1}	$2.3 \times 10^{+1}$
Ex8	1.2×10^{-2}	2.3×10^{-2}	3.0×10^{-2}	3.5×10^{-2}	6.8×10^{-0}
Ex9	2.1×10^{-2}	4.5×10^{-2}	5.6×10^{-2}	5.5×10^{-2}	$3.0 \times 10^{+1}$
Ex10	4.2×10^{-2}	2.7×10^{-1}	3.2×10^{-1}	2.1×10^{-1}	6.3×10^{-0}
Ex11	1.1×10^{-1}	9.0×10^{-1}	1.1×10^{-0}	7.6×10^{-1}	8.2×10^{-0}
Ex12	2.8×10^{-2}	2.2×10^{-1}	2.4×10^{-1}	8.3×10^{-2}	$3.0 \times 10^{+1}$
Ex13	3.2×10^{-0}	2.7×10^{-1}	3.4×10^{-1}	1.5×10^{-1}	$2.9 \times 10^{+1}$
Ex14	5.6×10^{-2}	2.1×10^{-1}	2.6×10^{-1}	1.3×10^{-1}	$1.7 \times 10^{+1}$
Ex15	5.7×10^{-2}	2.7×10^{-1}	3.3×10^{-1}	1.9×10^{-1}	$1.6 \times 10^{+1}$
Ex16	3.0×10^{-2}	3.3×10^{-1}	3.1×10^{-1}	1.7×10^{-1}	6.0×10^{-1}
Ex17	6.8×10^{-2}	2.7×10^{-1}	2.8×10^{-1}	1.5×10^{-1}	7.7×10^{-0}
Ex18	1.6×10^{-1}	1.1×10^{-0}	1.3×10^{-0}	2.9×10^{-1}	$9.0 \times 10^{+1}$
Ex19	1.6×10^{-1}	8.1×10^{-1}	1.2×10^{-0}	2.9×10^{-1}	$9.0 \times 10^{+1}$
Ex20	1.2×10^{-2}	1.5×10^{-2}	2.1×10^{-2}	2.9×10^{-2}	$6.0 \times 10^{+1}$
Ex21	1.2×10^{-2}	1.1×10^{-2}	2.1×10^{-2}	2.9×10^{-2}	$6.0 \times 10^{+1}$
Ex22	3.5×10^{-2}	7.9×10^{-2}	8.8×10^{-2}	6.4×10^{-2}	5.1×10^{-0}
Ex23	7.5×10^{-2}	3.6×10^{-1}	4.2×10^{-1}	2.0×10^{-1}	$1.9 \times 10^{+1}$
Ex24	1.5×10^{-1}	8.4×10^{-2}	1.0×10^{-1}	NA	5.4×10^{-0}
Ex25	1.1×10^{-1}	5.3×10^{-2}	7.0×10^{-2}	NA	3.7×10^{-0}

Table 6 reports CPU times in seconds. Many of the CPU times were small compared to accuracy of the clock used (accurate within a hundredth of a second). In order to accurately measure CPU time, shorter jobs were iterated several times within a single job run and CPU times were computed by dividing total CPU time by the number of iterations. Since a time sharing system was used, runs were made stand alone at night so that a particular job stream would not skew the CPU times reported.

Conclusions

When attempting to draw conclusions from the above data, it is important to keep in mind that the test polynomials used were designed to test for known specific potential problems that root solving routines, in general, have. Secondly, how one draws conclusions from the above data may depend on the specific application one wants to make of a root finding routine. For example, if only a few polynomials need to have their roots found, the CPU time is likely not to be very important whereas someone needing the roots of thousands of polynomials of reasonably high degree might be very concerned about efficiency. Clearly accuracy and efficiency play a joint role in evaluating performance. A generally inaccurate method is worth little even though it may be very efficient.

One way to compare the relative worth of the five algorithms is as follows. Let k be a positive integer and define a relative performance index $P_k, 0 \leq P_k \leq 1$, for each example problem, as

$$P_k = g_k \cdot (u_k/v_k),$$

where

(a) $g_k = \begin{cases} 1, & \text{if the average relative error } \leq 10^{-k}, \\ 0, & \text{if the average relative error } > 10^{-k}, \end{cases}$

(b) u_k is the smallest CPU time of all methods whose average relative error is $\leq 10^{-k}$ for the given example problem, and

(c) v_k is the CPU time of the method under consideration.

For example, if the average relative error for algorithms JIM, EM, EMB, LM and WM were $10^{-10}, 10^{-11}, 10^{-12}, 10^{-13}$ and 10^{-14}, and if the corresponding CPU times are 1, 2, 3, 4, and 5 seconds (respectively) and if $k = 12$, then the performance indices for these five algorithms are 1, 1/2, 1/3, 0 and 0, respectively. Thus, if an algorithm cannot achieve small enough average relative error, then it is given a zero performance index and the remaining algorithms are evaluated on the basis of relative CPU times. ($P_k = 0$ means poor relative performance; $P_k = 1$ means best relative performance.)

Tables 7 and 8 list the performance indices (rounded) for $k = 10$ and $k = 4$ respectively. The last row in each table, labelled "Total", is the sum of the performance indices for the algorithm listed above it. This "Total" will then give a performance rating of an algorithm over all examples used. Since Laguerre's Method (LM) was not applicable to Ex24 and Ex25, the "Aver" row is the average (rounded) performance rating, which should be used to compare the "performance" of the different methods. Notice the relative values of these averages are roughly the same for P_{10} and P_4.

Table 7. P_{10} **Performance numbers using average relative error**

	JTM	EM	EMB	LM	WM
Ex1	1.00	0.00	0.63	0.34	0.00
Ex2	1.00	0.00	0.50	0.34	0.00
Ex3	1.00	0.00	0.50	0.61	0.00
Ex4	1.00	0.45	0.39	0.37	0.00
Ex5	1.00	0.48	0.42	0.40	0.03
Ex6	0.26	0.83	0.71	1.00	0.01
Ex7	1.00	0.19	0.16	0.46	0.00
Ex8	1.00	0.52	0.40	0.34	0.02
Ex9	1.00	0.00	0.00	0.38	0.00
Ex10	0.00	0.00	0.00	0.00	0.00
Ex11	0.00	0.00	0.00	0.00	0.00
Ex12	0.00	0.00	0.00	0.00	0.00
Ex13	0.00	0.00	0.00	0.00	0.00
Ex14	1.00	0.00	0.22	0.43	0.03
Ex15	1.00	0.00	0.00	0.30	0.04
Ex16	0.00	0.00	0.00	0.00	0.00
Ex17	1.00	0.25	0.24	0.45	0.00
Ex18	1.00	0.15	0.12	0.55	0.00
Ex19	1.00	0.00	0.00	0.55	0.00
Ex20	1.00	0.80	0.57	0.41	0.00
Ex21	0.92	1.00	0.52	0.38	0.00
Ex22	1.00	0.44	0.40	0.55	0.07
Ex23	1.00	0.21	0.18	0.38	0.04
Ex24	0.00	0.00	0.00	NA	0.00
Ex25	0.00	0.00	0.00	NA	0.00
Total	17.18	5.32	5.96	8.24	0.24
Aver	0.69	0.21	0.24	0.36	0.01

Table 8. P_4 Performance numbers using average relative error

	JTM	EM	EMB	LM	WM
Ex1	1.00	0.00	0.63	0.34	0.00
Ex2	1.00	0.75	0.50	0.34	0.00
Ex3	1.00	0.00	0.50	0.61	0.00
Ex4	1.00	0.45	0.39	0.37	0.00
Ex5	1.00	0.48	0.42	0.40	0.03
Ex6	0.26	0.83	0.71	1.00	0.01
Ex7	1.00	0.19	0.16	0.46	0.00
Ex8	1.00	0.52	0.40	0.34	0.02
Ex9	1.00	0.47	0.38	0.38	0.00
Ex10	1.00	0.06	0.08	0.05	0.01
Ex11	0.00	0.00	0.00	0.00	0.00
Ex12	0.00	1.00	0.00	1.00	0.00
Ex13	0.05	0.00	0.44	1.00	0.01
Ex14	1.00	0.27	0.22	0.43	0.03
Ex15	1.00	0.21	0.17	0.30	0.04
Ex16	1.00	0.00	0.00	0.00	0.00
Ex17	1.00	0.00	0.00	0.00	0.00
Ex18	1.00	0.15	0.12	0.55	0.00
Ex19	1.00	0.00	0.12	0.55	0.00
Ex20	1.00	0.80	0.57	0.41	0.00
Ex21	0.92	1.00	0.52	0.38	0.00
Ex22	1.00	0.44	0.40	0.55	0.07
Ex23	1.00	0.21	0.18	0.38	0.04
Ex24	0.56	1.00	0.84	NA	0.02
Ex25	0.48	1.00	0.76	NA	0.01
Total	20.27	9.83	8.51	9.84	0.29
Aver	0.81	0.39	0.34	0.43	0.01

Table 9. P_{10} Performance numbers using maximum absolute error

	JTM	EM	EMB	LM	WM
Ex1	0.83	1.00	0.53	0.29	0.00
Ex2	1.00	0.75	0.50	0.34	0.00
Ex3	1.00	0.00	0.00	0.61	0.00
Ex4	1.00	0.45	0.39	0.37	0.00
Ex5	1.00	0.48	0.42	0.40	0.00
Ex6	0.26	0.08	0.07	1.00	0.00
Ex7	1.00	0.00	0.00	0.00	0.00
Ex8	1.00	0.52	0.40	0.34	0.00
Ex9	1.00	0.00	0.00	0.38	0.00
Ex10	0.00	0.00	0.00	0.00	0.00
Ex11	0.00	0.00	0.00	0.00	0.00
Ex12	0.00	0.00	0.00	0.00	0.00
Ex13	0.00	0.00	0.00	0.00	0.00
Ex14	1.00	0.27	0.22	0.43	0.00
Ex15	1.00	0.21	0.17	0.30	0.00
Ex16	0.00	0.00	0.00	0.00	0.00
Ex17	1.00	0.25	0.24	0.45	0.00
Ex18	1.00	0.00	0.00	0.55	0.00
Ex19	1.00	0.00	0.00	0.55	0.00
Ex20	1.00	0.80	0.57	0.41	0.00
Ex21	1.00	0.00	0.00	0.41	0.00
Ex22	1.00	0.44	0.40	0.55	0.01
Ex23	1.00	0.21	0.18	0.38	0.00
Ex24	0.00	0.00	0.00	NA	0.00
Ex25	0.00	0.00	0.00	NA	0.00
Total	17.09	5.46	4.09	7.76	0.01
Aver	0.68	0.22	0.16	0.34	0.00

Table 10. P_4 **Performance numbers using maximum absolute error**

	JTM	EM	EMB	LM	WM
Ex1	0.83	1.00	0.53	0.29	0.00
Ex2	1.00	0.75	0.50	0.34	0.00
Ex3	1.00	0.00	0.50	0.61	0.00
Ex4	1.00	0.45	0.39	0.37	0.00
Ex5	1.00	0.48	0.42	0.40	0.00
Ex6	0.26	0.08	0.07	1.00	0.00
Ex7	1.00	0.19	0.16	0.46	0.01
Ex8	1.00	0.52	0.40	0.34	0.00
Ex9	1.00	0.47	0.38	0.38	0.00
Ex10	1.00	0.16	0.13	0.20	0.01
Ex11	1.00	0.00	0.00	0.00	0.01
Ex12	0.00	1.00	0.92	0.00	0.00
Ex13	0.05	0.56	0.44	1.00	0.01
Ex14	1.00	0.27	0.22	0.43	0.00
Ex15	1.00	0.21	0.17	0.30	0.00
Ex16	1.00	0.00	0.00	0.00	0.00
Ex17	1.00	0.25	0.24	0.45	0.00
Ex18	1.00	0.15	0.12	0.55	0.00
Ex19	1.00	0.00	0.13	0.55	0.00
Ex20	1.00	0.80	0.57	0.41	0.00
Ex21	0.92	1.00	0.52	0.38	0.00
Ex22	1.00	0.44	0.40	0.55	0.01
Ex23	1.00	0.21	0.18	0.38	0.00
Ex24	0.56	1.00	0.84	NA	0.02
Ex25	0.48	1.00	0.76	NA	0.14
Total	21.10	10.99	8.99	9.39	0.21
Aver	0.84	0.44	0.36	0.41	0.01

To determine how much effect using "average relative error" has on the relative performance indices for these algorithms, we compute a performance index P_k exactly as defined above except that we now only use "maximum absolute error" instead of "average relative error". Tables 9 and 10 indicate the new values of the performance indices.

Whether one uses average relative error or maximum absolute error seems to have little effect on the overall relative performance of these five methods on this set of test polynomials. In all cases Wilf's method gave significantly superior performance over the other four algorithms whether or not one used average relative error or maximum absolute error, and whether or not one used an error cut-off of 10^{-4} or 10^{-10}.

References

[1] E. Hansen, M. Patrick and J. Rusnak, Some modifications of Laguerre's method, *BIT*, **17** (1977), 409–417.

[2] IMSL Library Reference Manual, Vol. I–III, International Mathematical & Statistical Libraries Inc., Houston, Texas, 1980.

[3] M.A. Jenkin and J.F. Traub, A three-stage algorithm for real polynomials using quadratic iteration, *SIAM J. Numer. Anal.*, **7** : 4 (1970), 545–566.

[4] M.A. Jenkin and J.F. Traub, A three-stage variable shift iteration for polynomial zeros and its relation to generalized Rayleigh iteration, *Numer. Math.*,**14** (1970), 252–263.

[5] M.A. Jenkin and J.F. Traub, Principles for testing polynomial zero finding programs, *ACM Trans. Math. Software*, I (1975), 26–34.

[6] L.W. Johnson and R.D. Riess, Numerical Analysis, Addison–Wesley, 1977.

[7] B. T. ZERPOL Smith, A zero finding algorithm for polynomials using Laguerre's method, Proc. of the 1967 Army Numerical Analysis Conference (ARO–D Report 67–3, Catalog No. AD–825 963).

[8] H.S. Wilf, A global bisection algorithm for computing the zeros of polynomials in the complex plane, *J. AMC*, **25** : 3 (1978), 415–420.