Contemporary Mathematics

A modified companion matrix method based on Newton polynomials

D. Calvetti, L. Reichel, and F. Sgallari

ABSTRACT. One of the most popular methods for computing the zeros of a polynomial in power form is to determine the eigenvalues of the associated companion matrix. This approach, which commonly is referred to as the companion matrix method, however, can yield poor accuracy. We demonstrate that when approximations of the zeros are available, their accuracy often can be improved by computing the eigenvalues of a modified companion matrix associated with the polynomial expressed in a basis of Newton polynomials defined by the available approximations of the zeros. The latter approximations can be determined in many ways, e.g., by the companion matrix method.

1. Introduction

The computation of the zeros $z_1, z_2, \ldots, z_n$ of a polynomial

$$\psi_n(z) = z^n + \alpha_{n-1}z^{n-1} + \ldots + \alpha_1 z + \alpha_0, \qquad \alpha_j \in \mathbb{C}, \tag{1}$$

is a fundamental task in scientific computation that arises in many problems in science and engineering. Many numerical methods have been proposed and analyzed for this problem; see, e.g., Henrici [**5**, Chapter 6] and Stoer and Bulirsch [**12**, Chapter 5] as well as the extensive bibliography provided by McNamee [**9**]. Among the most popular numerical schemes for polynomials of small to moderate degrees are the Jenkins-Traub method [**6**] and the companion matrix method. The latter method computes the zeros of the polynomial (1) as the eigenvalues of the companion matrix

$$C_n = \begin{bmatrix} 0 & & & & \cdots & 0 & -\alpha_0 \\ 1 & 0 & & & \cdots & 0 & -\alpha_1 \\ & 1 & 0 & & \cdots & 0 & -\alpha_2 \\ & & & & & \vdots & \vdots \\ & & & \ddots & \ddots & & \\ & & & & 1 & 0 & -\alpha_{n-2} \\ 0 & & & & & 1 & -\alpha_{n-1} \end{bmatrix} \in \mathbb{C}^{n\times n} \tag{2}$$

2000 *Mathematics Subject Classification.* Primary 65H05, 65F15, 26C10.
Key words and phrases. Companion matrix, Newton polynomial, zero-finder.
Research supported in part by NSF grants DMS 0107841 and 0107858.

by the QR-algorithm after balancing; see Edelman and Murakami [**2**] and Moler [**10**] for discussions. This method is used in the popular software package MATLAB [**8**] when calling the function `roots`. Goedecker [**3**] compared several implementations of the Jenkins-Traub method with the companion matrix method with regard to both accuracy and execution time for polynomials of small to moderate degree and found the latter approach to be competitive. We will throughout this paper refer to the companion matrix method as the CB method (where the letter C refers to the companion matrix and B to balancing).

However, it is fairly easy to find polynomials for which the CB method gives poor accuracy. Two modifications of the CB methods, referred to as the CBS and SHB methods, that for some polynomials can yield higher accuracy than the CB method were proposed in [**1**]. The CBS method first translates and scales the polynomial (1) to obtain a new polynomial

$$\hat{\psi}_n(\hat{z}) = \hat{z}^n + \hat{\alpha}_{n-2}z^{n-2} + \ldots + \hat{\alpha}_1\hat{z} + \hat{\alpha}_0, \qquad \hat{\alpha}_j \in \mathbb{C}, \tag{3}$$

with all zeros $\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n$ inside the unit disk and at least one zero fairly close to the unit circle, and such that $\sum_{j=1}^n \hat{z}_j = 0$. Thus, the zeros z_j of ψ_n and $\hat{z}_j$ of $\hat{\psi}_n$ are related according to

$$z_j = \sigma\hat{z}_j - \rho, \qquad 1 \le j \le n, \tag{4}$$

where the translation is given by $\rho = \alpha_{n-1}/n$ and the scaling factor σ is determined by using an inequality due to Ostrowski and the Schur-Cohn test; see [**1**] for details. Having determined the coefficients $\hat{\alpha}_j$ of $\hat{\psi}_n$, we compute the zeros $\hat{z}_j$ as the eigenvalues of the companion matrix associated with the polynomial $\hat{\psi}_n$. The desired zeros z_j are then determined by (4).

The SHB method is based on the observation that the polynomial (3) is a member of a family of monic Szegő polynomials $\phi_0, \phi_1, \ldots, \phi_n, \ldots$, where $\hat{\psi}_n(\hat{z}) = \phi_n(\hat{z})$. Szegő polynomials are orthogonal with respect to an inner product on the unit circle; see [**4**] for some of their properties and applications. The polynomial (3) can be expressed in terms of Szegő polynomials of lower degree

$$\hat{\psi}_n(\hat{z}) = \hat{z}\phi_{n-1}(\hat{z}) + \hat{\beta}_{n-1}\phi_{n-1}(\hat{z}) + \hat{\beta}_{n-2}\phi_{n-2}(\hat{z}) + \ldots + \hat{\beta}_1\phi_1(\hat{z}) + \hat{\beta}_0.$$

This expression and the recursion formulas for the Szegő polynomials define a Hessenberg matrix, whose eigenvalues are the zeros $\hat{z}_j$ of $\hat{\psi}_n$. The eigenvalues are computed by the QR-algorithm and the desired zeros z_j of ψ_n are obtained from (4).

This paper presents a new approach to improve the accuracy of available approximations $\hat{z}'_1, \hat{z}'_2, \ldots, \hat{z}'_n$ of the zeros of $\hat{\psi}_n$. These approximations may, for instance, have been computed by the CB, CBS or SHB methods. Let $z'_1, z'_2, \ldots, z'_n$ denote the associated approximations of the zeros of ψ_n, i.e., $z'_j = \sigma\hat{z}'_j - \rho$. Express the polynomial $\hat{\psi}_n$ in terms of a basis of Newton polynomials determined by the nodes $\hat{z}'_1, \hat{z}'_2, \ldots, \hat{z}'_n$, i.e.,

$$\hat{\psi}_n(\hat{z}) = \hat{\gamma}_0 + \hat{\gamma}_1(\hat{z} - \hat{z}'_1) + \hat{\gamma}_2(\hat{z} - \hat{z}'_1)(\hat{z} - \hat{z}'_2) + \ldots + \hat{\gamma}_n \prod_{j=1}^n (\hat{z} - \hat{z}'_j) \tag{5}$$

with $\hat{\gamma}_n = 1$. We define a modified companion matrix associated with the Newton form (5). The eigenvalues of this modified companion matrix are the zeros of $\hat{\psi}_n$. We compute the eigenvalues by the QR-algorithm after balancing of the matrix.

This defines the modified companion matrix (MCB) method. Let $\hat{z}_1'', \hat{z}_2'', \ldots, \hat{z}_n''$ denote the zeros computed by the MCB method, and let $z_1'', z_2'', \ldots, z_n''$ denote the associated approximations of the zeros of ψ_n, i.e., $z_j'' = \sigma \hat{z}_j'' - \rho$. Then, typically, z_j'' is a better approximation of z_j than z_j'.

Details of the MCB method are presented in Section 2, computed examples are discussed in Section 3, and concluding remarks can be found in Section 4.

2. The modified companion matrix method

Let $\hat{z}_1', \hat{z}_2', \ldots, \hat{z}_n'$ be points in the complex plane. In the computations presented in Section 3, the $\hat{z}_j'$ are computed approximations of zeros of the polynomial (3), however, for the discussion of the present section they may be arbitrary points in the finite complex plane.

We compute the coefficients $\hat{\gamma}_j$ of the representation (5) of the polynomial $\hat{\psi}_n$ in Newton form, given the representation (3) in power form and the nodes $\hat{z}_j'$. The computations are described by the following algorithm. The algorithm can be used to transform a general polynomial of degree n with leading coefficient $\hat{\alpha}_n$ to a polynomial in Newton form. In particular, the algorithm does not use that $\hat{\alpha}_n = 1$ and $\hat{\alpha}_{n-1} = 0$.

ALGORITHM 2.1 (Transformation from power basis to Newton basis).
Input: n, $\{\hat{\alpha}_k\}_{k=0}^n$, $\{\hat{z}_k'\}_{k=1}^n$;
Output: $\{\hat{\gamma}_k\}_{k=0}^n$;
$\hat{\gamma}_n := \hat{\alpha}_n$;
for $j := 1, 2, \ldots, n$ **do**
  $\hat{\gamma}_{n-j} := \hat{\alpha}_{n-j}$;
  for $k := 1, 2, \ldots, j$ **do**
    $\hat{\gamma}_{n-k} := \hat{\gamma}_{n-k} + \hat{\gamma}_{n-k+1} \hat{z}_{j-k+1}'$;
  end k;
end j; □

The computations of Algorithm 2.1 require $\mathcal{O}(n^2)$ arithmetic floating point operations.

In order to avoid large intermediate quantities, which may cause loss of accuracy during the computation with Algorithm 2.1, we Leja order the nodes $\hat{z}_j'$ before application of the algorithm. This ordering is defined by the following algorithm. The set $\mathbb{K}$ in the algorithm is the set $\{\hat{z}_j'\}_{j=1}^n$ of nodes, in arbitrary order, to be used in the Newton form. The algorithm assumes the nodes to be distinct. This restriction can easily be removed, see below.

ALGORITHM 2.2 (Leja ordering of nodes).
Input: *Set* $\mathbb{K}$ *of* n *distinct nodes;*
Output: *Leja ordered nodes* $\hat{z}_1', \hat{z}_2', \ldots, \hat{z}_n'$;
Let $\hat{z}_1'$ *be a node in* $\mathbb{K}$, *such that* $\hat{z}_1' = \max_{\hat{z} \in \mathbb{K}} |\hat{z}|$;
for $j := 2, 3, \ldots, n$ **do**
  Determine $\hat{z}_j' \in \mathbb{K}$, *so that*

$$\prod_{k=1}^{j-1} |\hat{z}_j' - \hat{z}_k'| = \max_{\hat{z} \in \mathbb{K}_\ell} \prod_{k=1}^{j-1} |\hat{z} - \hat{z}_k'| \ ;$$

end j; □

We refer to the ordering determined by Algorithm 2.2 as "Leja ordering," because when $\mathbb{K}$ is a compact continuum in the complex plane, the points determined by the algorithm are known as Leja points; see Leja [**7**] for a definition of Leja points and some of their properties. We have found experimentally that Leja ordering of the nodes before application of Algorithm 2.1 generally avoids that intermediate values of the coefficients $\hat{\gamma}_{n-k}$ computed during the execution of the algorithm are of large magnitude when the coefficients of the polynomial (3) are not. Large intermediate values may reduce the accuracy in the computed coefficients of the polynomial (3). Algorithm 2.2 can be implemented to require only $\mathcal{O}(n^2)$ arithmetic floating point operations.

When some nodes in the set $\mathbb{K}$ in Algorithm 2.2 are not distinct, then these nodes are ordered consecutively. Moreover, when the coefficients α_j of the polynomial (1) are real, the nodes $\hat{z}'_j$ are real or appear in complex conjugate pairs. We order distinct complex conjugate nodes consecutively.

Having determined the coefficients $\hat{\gamma}_j$ of the Newton form (5) by application of Algorithms 2.2 and 2.1, we form the modified companion matrix associated with the Newton form. It is given by

$$N_n = \begin{bmatrix} \hat{z}'_1 & & & & \cdots & 0 & -\hat{\gamma}_0 \\ 1 & \hat{z}'_2 & & & \cdots & 0 & -\hat{\gamma}_1 \\ & 1 & \hat{z}'_3 & & \cdots & 0 & -\hat{\gamma}_2 \\ & & & & & \vdots & \vdots \\ & & & \ddots & \ddots & & \\ & & & & 1 & \hat{z}'_{n-1} & -\hat{\gamma}_{n-2} \\ \text{\Large 0} & & & & & 1 & \hat{z}'_n - \hat{\gamma}_{n-1} \end{bmatrix} \in \mathbb{C}^{n\times n}. \tag{6}$$

We compute the eigenvalues of the matrix N_n by the QR-algorithm after balancing and denote the computed eigenvalues by $\hat{z}''_j$, $1 \le j \le n$. This defines the MCB method.

In our application of the MCB method, the nodes $\hat{z}'_j$ are approximations of zeros of $\hat{\psi}_n$ computed by the CB, CBS or SHB methods. Typically, the zeros $\hat{z}''_j$ computed by the MCB method are better approximations of the zeros $\hat{z}_j$ of $\hat{\psi}_n$ than the nodes $\hat{z}'_j$. We refer to the combined method that first determines approximations $\hat{z}'_j$ of the zeros $\hat{\psi}_n$ by the CB method, and then improves their accuracy by the MCB method as the CB+MCB method. The CBS+MCB and SHB+MCB methods are defined analogously.

Note that if the nodes $\hat{z}'_j$ are the exact zeros of the polynomial (3), then the coefficients $\hat{\gamma}_j$, when evaluated exactly, all vanish, and the matrix N_n is lower bidiagonal. Its eigenvalues are the exact zeros of the polynomial (3). If the nodes $\hat{z}'_j$ are close to the exact zeros, then the coefficients $\hat{\gamma}_j$ are of small magnitude and the matrix N_n is close to a lower bidiagonal matrix.

We may apply the MCB method recursively by using the approximations $\hat{z}''_j$ of the zeros obtained by the MCB method to define a new Newton basis and then applying the MCB method to compute new approximations $\hat{z}'''_j$ of the zeros. For instance, we may define the CB+MCB+MCB method in this manner. However, computational experience suggests that the latter method typically does not yield higher accuracy of the computed zeros than the CB+MCB method. Our experience with the CBS+MCB+MCB and SHB+MCB+MCB methods is similar.

TABLE 1. Comparison of zero finders applied to polynomials of degree 30 with zeros uniformly distributed in the square (7) with no pair of zeros closer than 0.2.

Method	Average Error	Smallest Error
SHB+MCB	$5.8 \cdot 10^{-5}$	97%
SHB	$6.2 \cdot 10^{-3}$	3%
CBS+MCB	$5.6 \cdot 10^{-5}$	97%
CBS	$1.8 \cdot 10^{-2}$	3%
CB+MCB	$1.6 \cdot 10^{-4}$	99%
CB	$4.8 \cdot 10^{-3}$	1%

3. Computed examples

This section presents computed examples that illustrate the performance of the CB+MCB, CBS+MCB and SHB+MCB methods. The computer programs used were all written in FORTRAN 77, and the numerical experiments were carried out on an HP9000/770 workstation in single-precision arithmetic, i.e., with approximately 7 significant decimal digits of accuracy. The eigenvalue problems were solved by single-precision subroutines from EISPACK [**11**].

In our experiments, we provided a set of n real or complex conjugate zeros of the polynomial ψ_n, defined by (1), and computed the coefficients α_j of the power basis representation by a recursion formula. The zeros were Leja ordered by Algorithm 2.2 in order to avoid unnecessary loss of accuracy; see the discussion in Section 2. The zeros were uniformly distributed in the square

$$\mathbb{S} = \{z \in \mathbb{C} : -1 \le \mathrm{Re}(z) \le 1, -1 \le \mathrm{Im}(z) \le 1\} \tag{7}$$

or in the unit disk

$$\mathbb{D} = \{z \in \mathbb{C} : |z| \le 1\}. \tag{8}$$

Since the zeros of ψ_n are real or appear in complex conjugate pairs, the matrices whose eigenvalues need to be computed in the CB, CBS and SHB methods are real and of upper Hessenberg form. We computed the eigenvalues of these matrices using the EISPACK subroutines `balanc` and `hqr`. The former implements a balancing scheme, the latter the QR algorithm. The modified companion matrix (6) has complex entries whenever ψ_n has non-real zeros. Its eigenvalues were computed by the EISPACK subroutine `comqr` after balancing by the subroutine `cbal`.

We measure the accuracy of the approximate zeros computed by the different methods by determining the distance between the set of exact zeros $\mathbb{Z}_n = \{z_j\}_{j=1}^n$ and the set of computed eigenvalues, say, $\mathbb{Z}''_n = \{z''_j\}_{j=1}^n$. The distance is defined by

$$\mathrm{distance}(\mathbb{Z}_n, \mathbb{Z}''_n) = \max\{\max_{z'' \in \mathbb{Z}''_n} ||z'' - \mathbb{Z}_n||, \max_{z \in \mathbb{Z}_n} ||z - \mathbb{Z}''_n||\}, \tag{9}$$

where $||z'' - \mathbb{Z}_n|| = \min_{z \in \mathbb{Z}_n} |z'' - z|$. We refer to $\mathrm{distance}(\mathbb{Z}_n, \mathbb{Z}''_n)$ as the error in the approximate zeros.

Example 3.1. We consider polynomials of degree 30, whose zeros are uniformly distributed in the square (7). The zeros are either real or appear in complex conjugate pairs, and the distance between each pair of zeros is required to be at least

TABLE 2. Comparison of zero finders applied to polynomials of degree 30 with zeros uniformly distributed in the square (7) with no pair of zeros closer than 0.2.

Error Quotient	Min	Max
error(SHB+MCB)/error(SHB)	$3.2 \cdot 10^{-4}$	4.6
error(CBS+MCB)/error(CBS)	$1.8 \cdot 10^{-4}$	1.0
error(CB+MCB)/error(CB)	$9.9 \cdot 10^{-4}$	1.7

TABLE 3. Comparison of zero finders applied to polynomials of degree 30 with zeros uniformly distributed in the square (7) with no pair of zeros closer than 0.1.

Method	Average Error	Smallest Error
SHB+MCB	$4.2 \cdot 10^{-3}$	93%
SHB	$2.4 \cdot 10^{-2}$	7%
CBS+MCB	$4.9 \cdot 10^{-3}$	98%
CBS	$4.6 \cdot 10^{-2}$	2%
CB+MCB	$1.8 \cdot 10^{-3}$	98%
CB	$1.2 \cdot 10^{-2}$	1%

0.2. The latter requirement is imposed in order to avoid severe ill-conditioning of the eigenvalue problems. For severely ill-conditioned eigenvalue problems the improvement in accuracy achieved by the MCB method is typically insignificant.

Table 1 shows the average error obtained by the different zero finders when applied to 100 polynomials of degree 30 generated as described above. The error in the polynomial zeros is defined according to (9). The SHB+MCB and CBS+MCB methods can be seen to give smaller errors than the other methods.

The MCB method reduces the error in the computed zeros of almost all polynomials. The last column of Table 1 shows for how many polynomials (out of 100) the MCB method reduces the error. For instance, the SHB+MCB method yields a smaller error than the SHB method for 97 of the 100 generated polynomials.

It is interesting to see how much the MCB method decreases or increases the error in available approximations of zeros. We therefore evaluate, for each polynomial, the quotient of the error in the computed zeros by the SHB and the SHB+MCB methods. The maximum and minimum of these quotients over the 100 polynomials in our experiment are displayed in Table 2. The table also shows the corresponding quotients associated with the CBS, CBS+MCB, CB and CB+MCB methods.

We remark that the errors reported in Tables 1-2 are for 100 consecutive polynomials determined by a random number generator. The entries of the tables vary somewhat with the initial seed of the generator, however, in all our experiments the SHB+MCB and CBS+MCB methods gave the smallest average errors in the computed zeros. □

Example 3.2. This example differs from Example 3.1 only in that the zeros are allowed to be closer. More precisely, the real or complex conjugate zeros of polynomials of degree 30 are uniformly distributed in the square (7), and the distance between each pair of zeros is required to be at least 0.1. Reducing the distance

TABLE 4. Comparison of zero finders applied to polynomials of degree 30 with zeros uniformly distributed in the square (7) with no pair of zeros closer than 0.1.

Error Quotient	Min	Max
error(SHB+MCB)/error(SHB)	$2.8 \cdot 10^{-4}$	4.0
error(CBS+MCB)/error(CBS)	$3.6 \cdot 10^{-5}$	1.3
error(CB+MCB)/error(CB)	$9.8 \cdot 10^{-5}$	1.1

TABLE 5. Comparison of zero finders applied to polynomials of degree 20 with zeros uniformly distributed in the square (7) with no pair of zeros closer than 0.2.

Method	Average Error	Smallest Error
SHB+MCB	$2.5 \cdot 10^{-5}$	72%
SHB	$4.4 \cdot 10^{-5}$	28%
CBS+MCB	$2.5 \cdot 10^{-5}$	73%
CBS	$1.4 \cdot 10^{-3}$	27%
CB+MCB	$1.2 \cdot 10^{-4}$	89%
CB	$3.0 \cdot 10^{-4}$	11%

TABLE 6. Comparison of zero finders applied to polynomials of degree 30 with zeros uniformly distributed in the unit disk (8) with no pair of zeros closer than 0.1.

Method	Average Error	Smallest Error
SHB+MCB	$4.5 \cdot 10^{-4}$	89%
SHB	$6.0 \cdot 10^{-3}$	11%
CBS+MCB	$4.4 \cdot 10^{-4}$	98%
CBS	$2.2 \cdot 10^{-2}$	2%
CB+MCB	$1.1 \cdot 10^{-3}$	100%
CB	$5.5 \cdot 10^{-2}$	0%

between zeros allows the eigenvalue problems to be more ill-conditioned. This has the effect that the average error in the computed zeros is larger than for Example 3.1. Table 3 shows the average error achieved by the different zero finders when applied to 100 polynomials and Table 4 displays error quotients. These tables are analogous to Tables 1 and 2, respectively. The CB+MCB method is seen to give the smallest average errors. □

Example 3.3. The polynomials generated in this example are of degree 20 with real or complex conjugate zeros uniformly distributed in the square (7). The distance between each pair of zeros is at least 0.2, as in Example 3.1. Table 5 shows the average errors achieved by the different methods. For all methods the average errors are smaller than in Table 1. The smallest average errors are achieved by the SHB+MCB and CBS+MCB methods. □

Example 3.4. The polynomials in this example are of degree 30 with real or complex conjugate zeros distributed uniformly in the unit disk (8). Similarly as in Example 3.2, the distance between each pair of zeros is at least 0.1. Table 6 shows the average errors achieved by the different methods. The smallest average errors are obtained by the SHB+MCB and CBS+MCB methods. □

4. Conclusion

Numerous numerical experiments, some of which have been presented in Section 3, indicate that the CB, CBS and SHB methods for computing zeros of polynomials can be enhanced by the MCB method. For many polynomials, the highest accuracy is achieved by combining the SHB or CBS methods with the MCB method.

References

[1] G. S. Ammar, D. Calvetti, W. B. Gragg, and L. Reichel, Polynomial zerofinders based on Szegő polynomials, J. Comput. Appl. Math., 127 (2001), pp. 1–16.
[2] A. Edelman and H. Murakami, Polynomial roots from companion matrix eigenvalues, Math. Comp., 64 (1995), pp. 763–776.
[3] S. Goedecker, Remark on algorithms to find roots of polynomials, SIAM J. Sci. Comput., 15 (1994), pp. 1059–1063.
[4] U. Grenander and G. Szegő, Toeplitz Forms and Applications, Chelsea, New York, 1984.
[5] P. Henrici, Applied and Computational Complex Analysis, vol. 1, Wiley, New York, 1974.
[6] M. A. Jenkins and J. F. Traub, A three-stage variable shift iteration for polynomial zeros and its relation to generalized Ritz iteration, Numer. Math., 14 (1970), pp. 252–263.
[7] F. Leja, Sur certaines suits liées aux ensemble plan et leur application à la representation conforme, Ann. Polon. Math., 4 (1957), pp. 8–13.
[8] Matlab, version 6.1, The MathWorks, Inc., Natick, MA, 2001.
[9] J. M. McNamee, An updated supplementary bibliography on roots of polynomials, J. Comput. Appl. Math., 110 (1999), pp. 305–306. This reference discusses the bibliography. The actual bibliography is available at the web site `http://www.elsevier.nl/locate/cam`.
[10] C. Moler, Cleve's corner: roots - of polynomials, that is, The MathWorks Newsletter, v. 5, n. 1 (1991), pp. 8–9.
[11] B. Smith, J. Boyle, Y. Ikebe, V. Klema, and C. Moler, Matrix Eigensystem Routines: EISPACK Guide, Springer, Berlin, 2nd ed., 1976.
[12] J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, 2nd ed., Springer, New York, 1993.

Department of Mathematics, Case Western Reserve University, Cleveland, OH 44106, USA
E-mail address: `dxc57@po.cwru.edu`

Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA
E-mail address: `reichel@mcs.kent.edu`

Dipartimento di Matematica, Universitá di Bologna, Piazza P.ta S. Donato 5, Bologna, Italy
E-mail address: `sgallari@dm.unibo.it`