Journal of Computational and Applied Mathematics 236 (2012) 3238–3255

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A fitting algorithm for real coefficient polynomial rooting

Peter Strobach

AST-Consulting Inc., Bahnsteig 6, 94133 Röhrnbach, Germany

ARTICLE INFO

Article history:
Received 2 June 2011
Received in revised form 14 December 2011

Keywords:
Polynomials
Root-finding
Convolution
Polynomial factorization
Polynomial fitting

ABSTRACT

A new algorithm for computing all roots of polynomials with real coefficients is introduced. The principle behind the new algorithm is a fitting of the convolution of two subsequences onto a given polynomial coefficient sequence. This concept is used in the initial stage of the algorithm for a recursive slicing of a given polynomial into degree-2 subpolynomials from which initial root estimates are computed in closed form. This concept is further used in a post-fitting stage where the initial root estimates are refined to high numerical accuracy. A reduction of absolute root errors by a factor of 100 compared to the famous Companion matrix eigenvalue method based on the unsymmetric QR algorithm is not uncommon. Detailed computer experiments validate our claims.

1. Introduction

Polynomial root-finding is one of the fundamental problems in computational mathematics with many applications in signal processing. Although considered classical, high rates of progress are still possible in this area, as we shall demonstrate in this article.

A large bibliography of root-finding algorithms can be found in [1,2]. A prominent research direction are Companion matrix methods for root finding. It is well known that the roots of a polynomial can be computed as the eigenvalues of the associated coefficient Companion matrix [3]. This way, the polynomial root-finding problem can be posed and solved customarily as an eigenvalue problem using the unsymmetric QR algorithm [3], for instance, using Lapack subroutine `dgeev.f`.

We can demonstrate that the numerical accuracy reached by the Companion matrix eigenvalue concept stays far below the numerical accuracy that can be reached in polynomial root finding. Moreover, the conventional unsymmetric QR algorithm assumes a dense matrix as input and cannot exploit the sparse structure of a Companion matrix. Consequently, an algorithm like `dgeev.f` is redundant both in terms of storage requirement and in terms of computational complexity when applied to a Companion matrix.

A great effort has gone into the development of fast QR algorithms for Companion matrices with the obligatory $O(n)$ storage and $O(n^2)$ computational complexity (see, for instance, the reference list in [4]). We found that these algorithms have not reached a competitive level yet because the performance in terms of absolute root errors is even worse than the level reached by `dgeev.f`.

This overall situation has motivated the development of a root-finding algorithm based on polynomial fitting. The basic step in this approach is the fitting of the convolution of two subsequences onto a given larger coefficient sequence. This gives rise to a nonlinear optimization problem which can be solved accurately and efficiently. The overall fitting rooter of this paper consists of a slicing or deflation-type preprocessor for coarse root estimation, and a subsequent post-fitting stage for root refinement based on a fitting of the coarse root estimates onto the entire clean input polynomial coefficient sequence.

E-mail address: peter_strobach@gmx.de.
URL: http://www.ast-consulting.net.

0377-0427/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2012.02.027

This paper is organized as follows: In Section 2, we introduce the basic theory and develop the slicing part of the algorithm. Several implementation details for this part of the algorithm are given in Section 3. Section 4 describes the post-fitting subalgorithm which determines the numerical accuracy of the final root estimates. Section 5 shows a collection of representative simulation results. Section 6 summarizes the conclusions.

2. The polynomial fitting approach

Suppose we have given a polynomial $A(z)$ of degree n

$$A(z) = z^n + a_1 z^{n-1} + a_2 z^{n-2} + \cdots + a_{n-1} z + a_n,$$

with real coefficients $\{a_k,\ k = 1, 2, \ldots, n\}$. We wish to compute the roots $\{z_k,\ k = 1, 2, \ldots, n\}$ of $A(z)$. This can be accomplished by computing factorizations $A(z) = B(z)C(z)$, where $B(z)$ is a subpolynomial of degree $n-2$ and $C(z)$ is a subpolynomial of degree 2. $A(z)$ and $C(z)$ have a common root pair which can be computed in closed form. Now $B(z)$ can be treated as a new (reduced) $A(z)$ amenable for a new factorization. This strategy of polynomial slicing or deflation can be repeated recursively until all roots of $A(z)$ have been found. Consequently, the main problem is the elementary factorization $A(z) = B(z)C(z)$. In this paper, we develop a fitting algorithm to accomplish this step. A fitting error polynomial $E(z)$ is hence introduced as follows:

$$E(z) = A(z) - B(z)C(z), \tag{1}$$

where

$$\begin{aligned} E(z) &= e_1 z^{n-1} + \cdots + e_{n-1} z + e_n, \\ B(z) &= z^{n-2} + b_1 z^{n-3} + \cdots + b_{n-3} z + b_{n-2}, \\ C(z) &= z^2 + c_1 z + c_2, \end{aligned}$$

with coefficients

$$\begin{aligned} \mathbf{a} &= [a_1, a_2, \ldots, a_{n-1}, a_n]^T, \\ \mathbf{e} &= [e_1, e_2, \ldots, e_{n-1}, e_n]^T, \\ \mathbf{b} &= [b_1, b_2, \ldots, b_{n-3}, b_{n-2}]^T, \\ \mathbf{c} &= [c_1, c_2]^T. \end{aligned}$$

Polynomial products are computed as convolutions of their coefficient sequences. Hence it is not difficult to verify that the z-domain representation of the fitting error (1) can be posed equivalently in the coefficient domain as follows:

$$\begin{bmatrix} 0 \\ e_1 \\ e_2 \\ \vdots \\ e_{n-1} \\ e_n \end{bmatrix} = \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \\ a_n \end{bmatrix} - \begin{bmatrix} 1 & & \\ b_1 & 1 & \\ b_2 & b_1 & 1 \\ \cdot & b_2 & b_1 \\ b_{n-2} & \cdot & b_2 \\ & b_{n-2} & \cdot \\ & & b_{n-2} \end{bmatrix} \begin{bmatrix} 1 \\ c_1 \\ c_2 \end{bmatrix}. \tag{2}$$

Band matrices $\mathbf{B}$ and $\mathbf{C}$ are introduced:

$$\mathbf{B} = \begin{bmatrix} 1 & \\ b_1 & 1 \\ b_2 & b_1 \\ \vdots & \vdots \\ b_{n-2} & b_{n-3} \\ & b_{n-2} \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} 1 & & & \\ c_1 & 1 & & \\ c_2 & c_1 & \ddots & \\ & c_2 & \ddots & 1 \\ & & \ddots & c_1 \\ & & & c_2 \end{bmatrix}, \tag{3}$$

where $\mathbf{B}$ is of dimension $n \times 2$ and $\mathbf{C}$ is of dimension $n \times (n-2)$. Convolutions commute. Consequently, error equation (2) can be expressed as follows:

$$\mathbf{e} = \mathbf{a}_b - \mathbf{Bc} = \mathbf{a}_c - \mathbf{Cb}, \tag{4}$$

where:

$$\begin{aligned} \mathbf{a}_b &= [a_1 - b_1, a_2 - b_2, \ldots, a_{n-2} - b_{n-2}, a_{n-1}, a_n]^T, \\ \mathbf{a}_c &= [a_1 - c_1, a_2 - c_2, a_3, \ldots, a_{n-1}, a_n]^T. \end{aligned}$$