Computers and Mathematics with Applications 72 (2016) 992–1001

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Accurate polynomial root-finding methods for symmetric tridiagonal matrix eigenproblems

L. Gemignani

Dipartimento di Informatica, Università di Pisa, Largo Bruno Pontecorvo, 3 - 56127 Pisa, Italy

ARTICLE INFO

Article history:
Received 14 December 2015
Received in revised form 6 June 2016
Accepted 8 June 2016
Available online 29 June 2016

Keywords:
Polynomial zeros
Tridiagonal matrix
Matrix eigenvalues
Three-term recurrence
Floating-point arithmetic
Accuracy

ABSTRACT

In this paper we consider the application of polynomial root-finding methods to the solution of the tridiagonal matrix eigenproblem. All considered solvers are based on evaluating the Newton correction. We show that the use of scaled three-term recurrence relations complemented with error free transformations yields some compensated schemes which significantly improve the accuracy of computed results at a modest increase in computational cost. Numerical experiments illustrate that under some restriction on the conditioning the novel iterations can approximate and/or refine the eigenvalues of a tridiagonal matrix with high relative accuracy.

1. Introduction

Polynomial root-finding algorithms can be applied for the solution of structured matrix eigenproblems. Most of the methods including the Newton method for eigenvalue refinement and the Ehrlich–Aberth iteration for simultaneous eigenvalue computation [1] need at each step to evaluate only the ratio $f(z)/f'(z)$ – generally referred as the Newton correction – between the value of the characteristic polynomial and of its first derivative. It is an immediate observation that the function value and the derivative might overflow/underflow while the ratio may still be a reasonable machine number. Numerically reliable polynomial methods should be able to exploit the structure of the matrix eigenproblem for the efficient and accurate evaluation of the Newton correction.

In this paper we focus on the symmetric tridiagonal eigenproblem. It is well known that in some cases the efficiency of the polynomial solver can be coupled with the high accuracy of the computed approximations. Theoretical and computational results have been already established in the literature for the important subclass of real symmetric tridiagonal matrices with zero diagonal entries. Such matrices arise naturally both in the framework of divide-and-conquer methods for bidiagonal singular value problems [2–4] and in the approximation theory for the computation of Gauss-type quadrature rules for symmetric weight functions [5,6]. Since symmetric tridiagonal matrices with zero diagonal specify their eigenvalues with high relative accuracy independently of their magnitudes [7] numerical methods can possibly compute these eigenvalues at the same relative accuracy they are determined by the input data. Relatively accurate polynomial zerofinders based on Laguerre's iteration are considered in [4,8] while *GR*-type matrix methods are devised in [7,9]. Similar results do not hold for a general symmetric tridiagonal whose entries do not determine its eigenvalues to high relative accuracy. Higher (multi)precision computation can be recommended to increase the accuracy of the computed approximations with some timing penalties.

E-mail address: l.gemignani@di.unipi.it.

http://dx.doi.org/10.1016/j.camwa.2016.06.009
0898-1221/

It is a classical result that the characteristic polynomial of a tridiagonal matrix together with its derivatives can efficiently be evaluated by using three-term recurrences [10]. Although such recurrences are computationally appealing and straightforward to implement in practice the resulting scheme can be prone to numerical difficulties. Due to overflow and underflow occurrences in real applications the computation needs to incorporate some normalization or scaling techniques [11]. In addition, the three-term sequence computation is backward stable but this does not imply any accuracy in the evaluation of the polynomial and its derivative and, a fortiori, of the corresponding Newton correction. According to the classical rule of thumb the (relative) forward error depends both on the (relative) backward error and the condition number. Wilkinson analyzed the three-term recurrence computation [10] by proving that the evaluation of the characteristic polynomial is relatively backward stable for points close to the origin. Nevertheless, quite commonly computing the determinant of a symmetric tridiagonal matrix is an ill-conditioned problem.

A similar situation also occurs with the Ruffini–Horner algorithm generally used to evaluate polynomials and incorporated in the multiprecision polynomial rootfinder MPSolve [12]. Though named for Paolo Ruffini (1765–1822) and William Horner (1786–1837), two European mathematicians who described it in the early 1800s, the Ruffini–Horner method had first appeared in mathematical texts of both Arab and Chinese medieval mathematicians [13] and then rediscovered in 1669 by Isaac Newton (see [14]). Very recently an accurate variant of the Ruffini–Horner scheme has been proposed in [15] which is capable to compute a result of the same quality as if computed using twice the working precision and then rounded to the working precision. This variant makes use of some modified algorithms – called error-free transformations in [16] – for evaluating the sum and the product of two floating point numbers introduced by Knuth [17] and Dekker–Veltkamp [18], respectively.

In this contribution we combine error-free transformations and scaled three-term recurrence relations to produce an efficient and accurate algorithm for evaluating both the characteristic polynomial of a symmetric tridiagonal matrix and its first derivative. By "accurate" we mean that the computed answers have relative errors as they were computed in twice the working precision. This means that we achieve full precision accuracy, apart from severely ill-conditioned computations, without timing penalties required by multiprecision environments. Then the algorithm is incorporated in the Newton method for testing purposes. By using a result in [19] the accurate computation of the characteristic polynomial implies that the same property holds for the approximation of the eigenvalues. Our numerical experience suggests that the resulting rootfinder is typically able to approximate matrix eigenvalues at high relative accuracy independently of their magnitude.

The paper is organized as follows. In Section 2 we introduce and analyze the compensated variants of the classical schemes making use of three-term recurrences for evaluating the characteristic polynomial, as well as its first derivative, of tridiagonal matrices. In Section 3 we illustrate the results of numerical experiments confirming the potential high relative accuracy of an eigenvalue refinement method based on the Newton method complemented with the compensated techniques for computing the function values. Finally, conclusion and future work are drawn in Section 4.

2. Accurate three-term recurrence computation

Let $T \in \mathbb{R}^{n\times n}$ be a symmetric unreduced tridiagonal matrix, i.e.,

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix}, \quad \beta_i, \alpha_i \in \mathbb{R},\ \beta_i \neq 0,\ 1 \leq i \leq n-1.$$

The characteristic polynomial

$$f_n(\lambda) = \det(\lambda I_n - T)$$

can be computed by the three-term recurrence relations

$$f_0(\lambda) = 1, \qquad f_1(\lambda) = \lambda - \alpha_1;$$
$$f_j(\lambda) = (\lambda - \alpha_j) f_{j-1}(\lambda) - \beta_{j-1}^2 f_{j-2}(\lambda), \quad j = 2, 3, \ldots, n.$$

By differentiating the relations we obtain a second recurrence for the evaluation of the first derivative $f_n'(\lambda)$, namely,

$$f_0'(\lambda) = 0, \qquad f_1'(\lambda) = 1;$$
$$f_j'(\lambda) = f_{j-1}(\lambda) + (\lambda - \alpha_j) f_{j-1}'(\lambda) - \beta_{j-1}^2 f_{j-2}'(\lambda), \quad j = 2, 3, \ldots, n.$$

The MatLab[1] function `evalpoly1` (**Algorithm 1**) [20] computes the function values $f_n(\lambda)$ and $f_n'(\lambda)$ for a given λ and returns the value of the Newton correction given by $r = \frac{f_n(\lambda)}{f_n'(\lambda)}$.

[1] Matlab is a registered trademark of The MathWorks, Inc.