Journal of Computational and Applied Mathematics 272 (2014) 449–467

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Structured matrix methods for the computation of multiple roots of a polynomial

Joab R. Winkler *

Department of Computer Science, The University of Sheffield, Regent Court, 211 Portobello, Sheffield S1 4DP, United Kingdom

HIGHLIGHTS

- It is shown that structured matrix methods allow multiple roots of a polynomial to be computed reliably.
- A geometric explanation, in terms of pejorative manifolds of a polynomial that has multiple roots, is provided and complements the numerical method.
- Structured matrix methods are used to perform polynomial deconvolution.
- A comparison of the results from the method described in the paper with the results from the suite of MATLAB programs MULTROOT is given.

ARTICLE INFO

Article history:
Received 30 December 2012
Received in revised form 24 July 2013

Keywords:
Roots of polynomials
Structured matrix methods

ABSTRACT

This paper considers the application of structured matrix methods for the computation of multiple roots of a polynomial. In particular, the given polynomial $f(y)$ is formed by the addition of noise to the coefficients of its exact form $\hat{f}(y)$, and the noise causes multiple roots of $\hat{f}(y)$ to break up into simple roots. It is shown that structured matrix methods enable the simple roots of $f(y)$ that originate from the same multiple root of $\hat{f}(y)$ to be 'sewn' together, which therefore allows the multiple roots of $\hat{f}(y)$ to be computed. The algorithm that achieves these results involves several greatest common divisor computations and polynomial deconvolutions, and special care is required for the implementation of these operations because they are ill-posed. Computational examples that demonstrate the theory are included, and the results are compared with the results from MULTROOT, which is a suite of MATLAB programs for the computation of multiple roots of a polynomial.

1. Introduction

The computation of the roots of a polynomial is a classical problem in mathematics and many methods have been developed, including the methods of bisection [1], Laguerre [2,3], Bairstow, Græffe and Müller [4], Horner [5], Jenkins and Traub [6], and Newton [7]. These methods have advantages and disadvantages with respect to speed, accuracy and convergence. For example, the methods of Müller and Newton may diverge, and the methods of Horner and Bairstow have good convergence properties, but all methods yield incorrect results when multiple roots are computed because these roots break up into simple roots. This has motivated the development of methods that are explicitly designed for the computation of multiple roots of a polynomial, that is, methods that do not cause the multiple roots of the polynomial to break up into simple roots [8–12]. The methods described in these papers require greatest common divisor (GCD) computations and

* Tel.: +44 114 222 1834; fax: +44 114 222 1810.
E-mail address: j.winkler@dcs.shef.ac.uk.

0377-0427/$ – see front matter
http://dx.doi.org/10.1016/j.cam.2013.08.032

polynomial deconvolutions, and the difficulty of computing multiple roots of a polynomial follows from the ill-posed nature of these operations. In particular, if two polynomials $\hat{f}(y)$ and $\hat{g}(y)$ have a non-constant GCD, their perturbed forms $f(y)$ and $g(y)$, respectively, are coprime with probability one (under very mild assumptions). Similarly, even if $\hat{g}(y)$ is an exact divisor of $\hat{f}(y)$, the deconvolution $f(y)/g(y)$ is a rational function and not a polynomial. It therefore follows that special care is required for the computational implementation of these operations, and it will be shown that structured matrix methods allow numerically robust solutions to be obtained. This numerical difficulty must be compared with the symbolic implementation of these operations on $\hat{f}(y)$ and $\hat{g}(y)$ using exact arithmetic, which does not present difficulties.

There also exist algorithms for the determination of real roots of a polynomial, for example, Sturm sequences, Descartes' rule of signs and continued fractions techniques, but these algorithms solve a problem that differs from the problem considered in this paper. Also, they fail to terminate when multiple roots are considered, unless additional information is provided.

The method whose implementation is described in this paper is similar to the method considered in [8–10], and it is based on the method described in [13], pages 65–68. The examples in [8–10] do not consider the effect of added noise, and thus an exact polynomial is specified and the errors in the computed results are due to roundoff error only. By contrast, the structured matrix methods used in this work allow the effects of noise to be considered, such that multiple roots of the exact polynomial $\hat{f}(y)$ are computed, even when an inexact (noisy) form $f(y)$ of $\hat{f}(y)$,

$$f(y) = \hat{f}(y) + \delta f(y), \tag{1}$$

is specified, where $\delta f(y)$ is the noise added to $\hat{f}(y)$.

The polynomial root solver MultRoot is a suite of Matlab programs for the computation of multiple roots of a polynomial in the presence of added noise [11,12], but it differs from the work described in this paper because the magnitude of the noise added to the coefficients of $\hat{f}(y)$ is required, such that a threshold on the singular values of the Sylvester matrix of $f(y)$ and its derivative $f^{(1)}(y)$ can be specified. By contrast, knowledge of this noise level is not required for the work described in this paper, which has practical advantages because the noise level may not be known, or it may only be known approximately, or the noise level may vary between the coefficients, such that a threshold cannot be specified *a priori*. In particular, computational experiments showed that if the signal-to-noise ratio is higher than a specified value, which is a function of the values and multiplicities of the roots, then the errors in the computed roots are small and knowledge of the noise level is not required. The dependence of the computed roots on these parameters has been observed by computational experiments, and not considered analytically, and thus only a qualitative statement, and not a quantitative statement, can be made. Also, computational experiments showed that the errors in the computed roots decrease as the noise level decreases, which is expected. If, however, the noise level is known, it can be used to verify the acceptability, or otherwise, of the computed roots.

The method whose computational implementation is considered in this paper is described in Section 2. This algorithm is explicitly designed for the computation of multiple roots, and its application to a polynomial, all of whose roots are simple, does not yield any advantages with respect to the methods discussed above. It is shown in Section 3 that the method described in this paper has a geometric interpretation based on the pejorative manifolds of a polynomial that has multiple roots, and that this allows an easier understanding of its numerical implementation [14]. Structured and unstructured condition numbers of a multiple root of a polynomial are discussed in Section 4, and it is shown they may differ by several orders of magnitude. This difference in magnitude provides the motivation for the method that implements the polynomial root solver described in this paper, and it forms a clear distinction between this root solver and the root solvers reviewed above.

The method whose implementation is described in this paper requires that polynomial deconvolutions of the form $h_i(y) = p_{i-1}(y)/p_i(y)$ be considered. It is clear that the ith and $(i+1)$th deconvolutions are coupled, and it is therefore advantageous to consider these deconvolutions simultaneously, such that the coupling is explicitly included in the formulation of the problem, rather than consider them as decoupled (independent) deconvolutions. It is shown in Section 5 that structured matrix methods can be used for the computation of these coupled deconvolutions. Section 6 contains examples of the polynomial root solver described in this paper, and the results are compared with the results obtained from the polynomial root solver MultRoot. Section 7 contains a summary of the paper.

This paper is a continuation of the work in [15], and it is shown in this paper that:

- Structured matrix methods allow multiple roots of an exact polynomial $\hat{f}(y)$ to be computed when an inexact form $f(y)$ of $\hat{f}(y)$ is given.
- The polynomial root solver used to compute multiple roots of $\hat{f}(y)$ when $f(y)$ is specified can be interpreted in terms of the pejorative manifolds of $\hat{f}(y)$. The equations that define the pejorative manifolds of a third order polynomial are derived.
- Structured and unstructured condition numbers of a multiple root of a polynomial are considered. It is shown that they may differ by several orders of magnitude, and that this difference increases as the multiplicity of the root increases.
- A structured matrix method can be used to perform polynomial deconvolution, and it is shown that this method is better than the method of least squares. Specifically, the structured matrix method returns the coefficients of an exact