February 1, 2007

A Method for Finding Real and Complex Roots of Real Polynomials With Multiple Roots

by C. Bond, ©2002

1 Background

A review of the literature on polynomial root (zero) finders reveals methods attributed to Laguerre, Newton, Graëffe, Rutishauser, Bairstow, Lin, Müller, and many others. Recent contributions to the art include methods by Brent and the well known, robust method of Jenkins and Traub. This long list of distinguished contributors underscores the importance of the problem, and the range of solution methods. Thanks to the fundamental theorem of algebra, it is known that an nth order polynomial has exactly n roots, not necessarily distinct, and the locations of those roots are of great interest in many branches of applied science.

Solution methods fall into one of several classes: those that focus on finding a real root of a polynomial, those that find all real roots of real polynomials, those that find one or more complex roots of real polynomials, and those that find roots of complex polynomials. Our focus is on finding all real or complex roots of real polynomials. This class of problems is the most important one in electrical and mechanical engineering and finds applications in Laplace transform theory.

Experienced numerical analysts know that polynomial root finding is, all too often, a tentative process. Any computer method using finite precision is guaranteed to fail for polynomials of sufficiently high degree. Pathological polynomials with tightly clustered roots or roots with large dynamic range are also problematic.

Nevertheless, there is reason to be optimistic for the kinds of polynomials normally encountered in practice. The solution of these polynomials is the subject of this paper.

2 Pitfalls in Finding Individual Polynomial Roots

It will be assumed that the reader is familiar with one or more root finding methods and has had some experience with difficult cases. Consult a text on numerical analysis if the subject matter is unfamiliar.

2.1 Distinct Roots

Generally, root finders have fewer problems when the roots are widely spaced relative to the machine epsilon. In fact, spacings exceeding machine epsilon by many orders of magnitude are sometimes necessary to assure accurate root estimation. [1] On the other hand, when roots are closely spaced, roots finders begin to lose accuracy in separating one from the other. For that reason, moving the problem to a platform which supports extended precision (smaller machine epsilon) may be necessary. Even then, when two or more roots coincide it is impossible to separate them and a specific, and common, set of problems arise which apply to platforms of any precision.

The major cause of difficulty in extracting multiple roots is that most polynomial root finders attempt to converge on a zero by implicitly or explicitly using the polynomial slope in the vicinity of the zero. However, at a root with multiplicity greater than one, the slope is zero at the root. This complicates the convergence process and for roots with high multiplicity may make acceptable convergence impossible.

An example might help clarify the situation. Using the product of factors form a polynomial, let

$$P = (x - a)(x - b)(x - c)^2$$

[1] Graëffe's method transforms one polynomial into another whose roots are spread further apart. This process is repeated until the largest root is widely separated from its nearest neighbor, at which point it is easy to estimate its value from the coefficient of the term with the second largest exponent.

be a 4th order polynomial in x with a double root at c. Then

$$\begin{aligned} P' &= (x-b)(x-c)^2 + (x-a)(x-c)^2 + 2(x-c)(x-a)(x-b) \\ &= (x-c)\left((x-b)(x-c) + (x-a)(x-c) + 2(x-a)(x-b)\right) \end{aligned} \quad (1)$$

which is zero at c. Hence, the both P and P' are zero at c.

Any method which uses Newton type iterations will have difficulty with multiple roots. That includes Bairstow's and Lin's methods. Müller's method has difficulties for a related reason. Specifically, it involves curve fitting a parabola to the polynomial near a root, and the parabola becomes pathologically extended at multiple roots. Note that a naïve implementation of Graëffe's method fails completely when all roots are the same, because they cannot be separated. With finite epsilon, the question: When are two very closely spaced roots considered identical? confronts (and confounds) the analyst.

2.2 Initial Estimates

Techniques which locate roots by iterative methods require some starting estimate to begin the process. If the estimate happens to be closer to a root which is different from the desired one, the process may converge to the wrong one. In the absence of sufficient prior knowledge of the root locations, this is always a possibility.

If no root estimate is available it is a common practice to start the iteration with a value of zero. The rationale for starting at zero, even after all zero roots have been removed, is that convergence to the smallest root is expected, and many texts on numerical analysis recommend starting there. Methods that attempt to find all roots will generally try to find them in order from smallest (in magnitude) to largest so that the process of deflation will preserve the greatest relative accuracy.[2] But note that Graëffe's method uses a strategy which starts with the largest root.

A pitfall of starting the search from zero occurs in the cases where the linear

[2] If the largest root is found first and divided out of the original polynomial, the accuracy of later determination of smaller roots may be seriously degraded by loss of precision.

term of the polynomial has a zero coefficient. Newton type methods use the slope at each root estimate to calculate the improved estimate. But the slope of a polynomial with no linear term is zero at zero, as can be seen by differentiating the polynomial and evaluating there. Hence, in this particular case, the method fails to converge.

2.3 Deflation

In order to locate all roots of a polynomial, it is necessary to find a strategy for converging to each root. This is sometimes difficult or impossible when, for example, one root is between two other closely spaced roots. The problem can be avoided by dividing each newly found root out of the polynomial, so only the other roots remain. Eventually all roots will be found, but if there are significant errors in the accepted root values, those errors are propagated through the deflation process. Subsequent roots will be progressively degraded in this case.

The best defense against loss of precision due to deflation, is to iterate each root estimate back into the original polynomial. Unfortunately, for closely spaced roots, the refinement process may actually fail to converge, complicating the process considerably or defeating it altogether.

It is clear that a strategy involving finding a root followed by deflation will preserve maximum accuracy if the roots are found in order from smallest to largest. But multiple roots again present a special problem, since they are all equal and any error in finding one will clearly degrade the accuracy of the others.

3 Properties of the Derivative Polynomial

Surprisingly, the derivative polynomial which is the source of problems with multiple roots, provides a key to an elegant solution for finding those roots. Recall Equation (1), which showed that the derivative of a polynomial containing multiple roots also contains that root. However, the order of the multiple root in the derivative polynomial is reduced by one. In fact, all roots with multiplicity greater than one are reduced by one with a single differentiation. A second differentiation will reduce the order by one again. The derivative of a polynomial with distinct roots does not contain any of those roots, as the new zeros correspond to peaks of the original polynomial. Hence, if we have an approximation to a root (or quadratic factor) which appears to have multiplicity greater than one (or two), we can refine the estimate in the derivative polynomial.

Since polynomial differentiation is a stable process, involving only multiplication by integers and addition, there is no danger of loss of accuracy in repeated differentiation. It is therefore possible to recursively differentiate a polynomial until any multiple root becomes a distinct root with high accuracy.

Although it is well known that the quotient of the original polynomial divided by the derivative polynomial reduces all multiple roots to unity order, a solver based on this approach is questionable. The reason is that the division can result in an unacceptable loss of accuracy.

In the following section, a new method for finding all real and complex roots of real polynomials based on recursive differentiation is disclosed.

4 Solution Algorithm

The proposed solution strategy can be expressed as an algorithm in pseudocode, but some explanation is necessary. The plan is to attempt to recover roots accurately using Bairstow's method, so that complex numbers are not required. In Bairstow's method, quadratic factors are found using a Newton-type iteration.

If convergence problems occur in the process of recovering a quadratic, the best quadratic estimate found is used to iterate in the derivative polynomial. Assuming the convergence problem was due to multiple roots ($r > 2$), the convergence should be improved. If convergence is still uncertain, a second differentiation is performed and the current best estimate is iterated on the new lower order polynomial. Eventually, the sought for root will be found with acceptable convergence behavior, or the polynomial will be differentiated until the multiple root is no longer present in the derivative polynomial.

Note that if the coefficient of the linear term of a polynomial is zero, the differentiated polynomial will have a root at zero. In this case the zero should be divided out by reducing the power of the variable by one for each term.

The various cases can be distinguished by observing the remainders on attempted deflation of the polynomial by the root estimate. As the estimate improves, the remainder approaches zero. If the root is not present the remainder will suddenly increase dramatically, in which case we revert to the previously found best estimate.

A pseudocode algorithm which embodies the above strategy is shown below.

1. Call Bairstow's method with *small* initial estimate.
2. If convergence was satisfactory, goto 10.
3. Let D = P'.
4. If the constant term is zero, shift all coefficients down reducing order by 1.
5. If the order of D is 2, estimate is D: goto 10.
6. Call Bairstow's method on D using estimate.
7. If convergence was satisfactory, goto 10.
8. If wrong root, restore D and previous estimate, goto 10.
9. Goto 3.
10. Deflate polynomial using estimate and save quadratic.
11. Replace original polynomial with reduced polynomial.
12. If the order of the reduced polynomial is greater than 2, goto 1.
13. Get remaining root(s) and return to caller.

5 Conclusion

A new method for extracting all real and complex roots from a real polynomial in the presence of multiple roots has been disclosed. The reason for convergence problems with multiple roots has been explained and a solution strategy has been described. A pseudocode algorithm which implements the solution has been provided.

Although the recursive refinement of multiple roots has been demonstrated using Bairstow's method, the same strategy can also be implemented with Müller's method, Newton's method or any other method for individual roots if complex numbers are supported.

The remaining problems in implementing the method involve *fine tuning* of the convergence evaluation metrics. This is not a trivial problem, but it is tractable, as is demonstrated by actual code available from the author.

5.1 Caveats

The proposed method is not and cannot be guaranteed to succeed in all cases. It is easy to construct polynomials which can defeat any root finder.[3] Nevertheless, it performs well with polynomials which cause other methods to fail and may earn a trusted place in the researchers tool box.

No formula is known for deriving optimal error metrics. It may be necessary to 'jiggle the handle' to get the kind of performance required for specific classes of polynomials.

As noted earlier, in extreme cases the only option is to employ extended precision.

[3]Polynomials of very high degree, or with roots which differ in magnitude by many orders of magnitude are problematic, as are polynomials with tight clusters of roots.