ROOTS OF POLYNOMIALS BY NEWTON'S ITERATION

by

David R. Beaucage

Department of Computer Science
Livingston College
Rutgers University
The State University of New Jersey
New Brunswick, New Jersey 08903

Department of Computer Science Technical Report #12

October, 1971

Roots of polynomials by Newton's iteration

David R. Beaucage

ABSTRACT. An iterative method is described which finds all the roots of a square-free polynomial at once, using the original coefficients in each step (thus minimizing roundoff error). Since the iteration is basically Newton's, the usual convergence criteria and quadratic-convergence behavior obtain. Except for the facts about polynomial coefficients and contraction mappings, the paper is self-contained.

KEYWORDS: roots, polynomial, iteration.

If $p(x) \in \mathcal{C}[x]$ is a monic polynomial of degree $n \geq 1$,

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + x^n,$$

the coefficients are (±) elementary symmetric functions of the roots:

$$a_n = s_0^n(r) \equiv 1,$$

$$a_{n-1} = -s_1^n(r) = -r_1 - r_2 - \dots - r_n,$$

$$a_{n-2} = s_2^n(r) = r_1r_2 + r_1r_3 + \dots + r_{n-1}r_n,$$

$$\dots\dots\dots\dots\dots\dots,$$

$$a_{n-i} = (-1)^i\, s_i^n(r) = (-1)^i \text{ (the sum of all products of i roots)},$$

$$\dots\dots\dots\dots\dots\dots,$$

$$a_0 = (-1)^n\, s_n^n(r) = (-1)^n\, r_1r_2\dots r_n,$$

where $r \in \mathcal{C}^n$ is any (column) vector of roots of $p(x)$, $r = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}$.

(There are at most n! such points $r \in \mathcal{C}^n$.)

If $\bar{z} \in \mathcal{C}^n$ is the vector $z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}$, let b_{n-i} be the coefficient of x^{n-i} in the polynomial $q(x) = (x-z_1)(x-z_2)\dots(x-z_n)$ (then $b_{n-i} = (-1)^i s_i^n(z)$). Define a function $f\colon \mathcal{C}^n \to \mathcal{C}^n$ by the formula

$$f\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} = \begin{pmatrix} a_{n-1} - b_{n-1} \\ a_{n-2} - b_{n-2} \\ \dots\dots\dots \\ a_0 - b_0 \end{pmatrix}.$$

Then finding all the roots of $p(x) = 0$ is equivalent to finding a point $r \in \mathcal{C}^n$ such that $f(r) = 0$; i.e., finding a single root of $f(z) = 0$.

Newton's iteration produces, from an arbitrary initial guess $z(0) = \begin{pmatrix} z_1(0) \\ z_2(0) \\ \vdots \\ z_n(0) \end{pmatrix}$, a sequence $\{z(k)\}$ by the algorithm

$$z(k+1) = z(k) - (f'(z(k)))^{-1} f(z(k)). \tag{1}$$

This sequence will converge to a root r of $f(z) = 0$ from any $z(0)$ near enough to r, as long as $f'(r)$ is nonsingular. To see this, consider the function $g\colon \mathcal{C}^n \to \mathcal{C}^n$ given by $g(z) = z - C\,f(z)$, where C is any $n\times n$ matrix with smooth entries. $g'(z) = I - (C'\,f(z) + C\,f'(z))$; in case C is the inverse of $f'(z)$, its entries are smooth (as we shall see) and

$$g'(z) = I - (C'\,f(z) + I) = -\,C'\,f(z),$$

where C' is the row-vector with entries the matrices $\frac{\partial C}{\partial z_j}$. Since C is smooth, and r is in its domain because $f'(r)$ is nonsingular, all the entries in C' are finite at r; therefore $g'(r) = -\,C'\,f(r) = 0$. Thus $|g'(r)| = 0$, and, since the norm (we are using the spectral radius) is continuous, for any $\varepsilon > 0$ there is a neighborhood N of r in which

$|g'(z)| < \varepsilon$. Then for some $\theta \in N$ we have $|g(z) - g(r)| = |g'(\theta)(z-r)| < \varepsilon|z-r|$; if we choose $\varepsilon < 1$, g is a contraction mapping, hence has a unique fixed point in N, the limit of the sequence produced by repeated application of g. I.e., (1) converges.

It remains to prove that $f'(r)$ is nonsingular if $p(x)$ is square-free, i.e., has no multiple roots. But to apply (1) in practice, it also appears necessary to compute $C = (f'(z))^{-1}$ at each iteration, a problem at best comparable with the original one. However, the following theorem removes this difficulty.

THEOREM. The i^{th} component of $z(k+1)$ in (1) is given by

$$z_i(k) - p(z_i(k))/q'(z_i(k)). \tag{2}$$

Before proving the theorem we shall prove several lemmas. First, however, we make some preliminary remarks. Notice that (2) is actually simpler than it looks, because

$$q'(z_i) = (z_i-z_1)(z_i-z_2)\ldots(z_i-z_{i-1})(z_i-z_{i+1})\ldots(z_i-z_n).$$

In fact, the procedure

```
ROOTS: PROCEDURE(A,N); DECLARE ((A,Z,D,P)(N),O) COMPLEX BINARY(53);
       O = 6.28318531I/N; Z(1) = 1;
       DO I=2 TO N; Z(I) = EXP(O*(I-1)); END;
       ON ERROR GO TO EXIT;
 LOOP: DO I=1 TO N; D = Z(I)-Z; D(I)=1; P(I) = POLY(A,Z(I))/PROD(D); END;
       Z = Z-P; PUT SKIP EDIT(Z)(F(10,5)); GO TO LOOP;
 EXIT: RETURN; END ROOTS;
```

is all that is required to convert coefficients into roots.

Note also that, as long as z is close enough to a root-vector r and all the roots $r_1, r_2, \ldots, r_n$ are distinct, $q'(z_i) \neq 0$ for each i. Further notice that, as one would hope, the correction term $-p(z_i)/q'(z_i)$ is zero at a root of $p(x)$, so we do not wander away from a root once found while seeking the others. Finally, we remark that (2) may be adapted (although ROOTS does not do this) to the method of successive displacements, since $q'(z_i)$ depends on previous z_j's $(j < i)$.

To present the lemmas requires some notation. Let

$$q_j(x) = (x-z_1)(x-z_2)\ldots(x-z_{j-1})(x-z_{j+1})\ldots(x-z_n)$$

(thus $q_j(z_j) = q'(z_j)$, and $q_j(z_i) = 0$ if $i \neq j$). We denote the coefficient of x^{n-i} in $q_j(x)$ by $d_{n-i}(j)$. Also, we denote by $s_i^{n-1}(\hat{z}_j)$ the i^{th} elementary symmetric function on the n-1 numbers $z_1, z_2, \ldots, z_{j-1}, z_{j+1}, \ldots, z_n$. Then, just as in p(x), $d_{n-i}(j) = (-1)^{i-1} s_{i-1}^{n-1}(\hat{z}_j)$.

LEMMA 1. $$\frac{\partial s_i^n(z)}{\partial z_j} = s_{i-1}^{n-1}(\hat{z}_j).$$

PROOF: This follows from the fact (which is evident from the definitions) that $s_i^n(z) = s_{i-1}^{n-1}(\hat{z}_j)\, z_j + s_i^{n-1}(\hat{z}_j)$. □

Now f'(z) has in its i^{th} row and j^{th} column the entry $-\frac{\partial b_{n-i}}{\partial z_j}$ $= -(-1)^i \frac{\partial s_i^n(z)}{\partial z_j} = (-1)^{i-1} s_{i-1}^{n-1}(\hat{z}_j) = d_{n-i}(j)$. Thus we have proved

LEMMA 2.

$$f'(z) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ d_{n-2}(1) & d_{n-2}(2) & \cdots & d_{n-2}(n) \\ d_{n-3}(1) & d_{n-3}(2) & \cdots & d_{n-3}(n) \\ \cdots & \cdots & \cdots & \cdots \\ d_0(1) & d_0(2) & \cdots & d_0(n) \end{pmatrix}. \square$$

LEMMA 3. Let $C = (c_{ij})$, where $c_{ij} = z_i^{n-j}/q'(z_i)$. Then C is the inverse of f'(z).

PROOF: Let $\bar{C}\, f'(z) = P = (p_{ij})$. Then

$$p_{ij} = \frac{1}{q'(z_i)} \sum_{k=1}^{n} z_i^{n-k}\, d_{n-k}(j) = \frac{q_j(z_i)}{q_i(z_i)} = \delta_{ij}. \;\Box$$

Remark 1. We now see that, as promised, when C exists its entries are smooth: in fact, c_{ij} is infinitely differentiable, because all its partial derivatives will be rational functions with denominator a power of $q'(z_i) \neq 0$.

Remark 2. Clearly, some $q'(r_i) = 0$ iff $p(x)$ has multiple roots. Therefore, if $p(x)$ is square-free, $f'(r)$ is nonsingular, and conversely. Thus (1) converges if $p(x)$ is square-free.

Remark 3. If $p(x)$ has real coefficients and $z(0)$ is real, the method will converge only if all the roots of p are real (and, of course, distinct). (To avoid this, ROOTS takes $z(0)$ with components equally spaced around the unit circle.) The proximate cause of failure is that $z(k)$ is trapped in the real subspace for each k, since (2) involves only rational operations; the projections onto the real subspace of a complex conjugate pair of roots $r_j = a+bi$ and $r_m = a-bi$ coincide at a; and the iteration blows up as $z_j(k)$ and $z_m(k)$ both approach a.

PROOF OF THE THEOREM: The i^{th} component of $z - C\, f(z)$ may now be computed to be

$$z_i - \sum_{j=1}^{n} c_{ij}(a_{n-j} - b_{n-j}) = z_i - \sum_{j=1}^{n} \frac{z_i^{n-j}}{q'(z_i)}(a_{n-j} - b_{n-j})$$

$$= z_i - \frac{1}{q'(z_i)}\left((p(z_i) - z_i^n) - (q(z_i) - z_i^n)\right)$$

$$= z_i - (p(z_i) - q(z_i))/q'(z_i) = z_i - p(z_i)/q'(z_i),$$

since z_i is a root of $q(x) = 0$. $\Box$

Remark 4. Duplicate roots may be removed from any polynomial p by rational operations: they all live in GCD(p,p'), which may be found by, e.g., Collins' version of the Euclidean algorithm, and factored out of p, leaving a square-free quotient. Thus our method is more widely applicable than may at first appear.

Remark 5. (2) may readily be used to refine roots found by some other method (e.g., Bairstow-Hitchcock). Its advantage in this regard is similar to the advantage of iterative methods for solving linear systems over algorithmic methods: roundoff error is not cumulative, but appears only as a result of each single step. Moreover, each step uses all the original coefficients of p(x) as given, in contrast to methods which extract one or two roots at a time.

Rutgers University

October 25, 1971