Roots of Polynomials

So far we have looked at methods for finding roots of arbitrary functions. Now let us specialize these to polynomials.

Applications: Why polynomials?

- We already saw the use of polynomials in approximation/interpolation. Finding their roots is a natural next step (we might have an approximate representation of a function whose roots we need, say, to ensure safety of a trajectory).

- Systems of multivariate polynomial equations have seen much attention recently, particularly in motion planning, and to some extent in grasping and in machine vision, "Exact" algebraic solutions (i.e., root bracketing) are used to find roots. (cf Canny, Theory of the Reals, etc.)

We will only look at polynomials in one variable.
(Side note: Often, the solution of multi-variate polynomials is transformed into a problem requiring the solution of a single-variate polynomial.)

Reminder:
- Closed-form formulas exist for finding the roots of polynomials of degree 1, 2, 3, & 4.
- It is impossible to have a general formula for degree ≥ 5.
- Even degree 3 & 4 are somewhat unstable numerically.

Some facts

Suppose we write a polynomial of degree n as:

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \quad , \text{with } a_n \neq 0$$

① p has n real or complex roots, counting multiplicities.

② If the coefficients $\{a_i\}$ are all real, then the complex roots occur in conjugate pairs.

③ Descartes' rule of sign

let v = number of variations in the sign of the coefficients $a_n, \ldots, a_0$.
(ignore coefficients that are zero)

let n_p = number of real positive zeros

Then: (i) $n_p \leq v$
(ii) $v - n_p$ is an even integer (possibly zero)

Similarly, the number of real negative zeros of $p(x)$ is related to the number of sign changes in the coefficients of $p(-x)$.

<u>Ex</u> $p(x) = x^4 + 2x^2 - x - 1$

Then $v = 1$, so n_p is either zero or 1 by (i).
But, by (ii) $v - n_p$ must be even. Hence $n_p = 1$.

Now look at $p(-x) = x^4 + 2x^2 + x - 1$
Again, this has one variation in sign, so $n_n = 1$ as well.

In short, simply by looking at the coefficients we know that $p(x)$ has:

1 positive real root
1 negative real root
2 complex roots, as a conjugate pair.

④ <u>Bounds</u> (generalization of root bracketing from the real line to the complex plane)

$p(x)$ has at least one (real or complex) root inside the circle of radius ρ about the origin, where

$$\rho = \min \left\{ n \frac{|a_0|}{|a_1|} , \sqrt[n]{\frac{|a_0|}{|a_n|}} \right\}$$

⑤ <u>More bounds</u> let $r = 1 + \max_{0 \le k \le n-1} \left| \frac{a_k}{a_n} \right|$

Then all zeros of $p(x)$ lie in the region $\{ z \mid |z| \le r \}$.

⑥ A theorem of Cauchy's

Given $p(x)$, define

$$P(x) = |a_n| x^n - |a_{n-1}| x^{n-1} - \cdots - |a_0|$$

$$Q(x) = |a_n| x^n + |a_{n-1}| x^{n-1} + \cdots + |a_1| x - |a_0|$$

By Descartes, $P(x)$ has exactly one real positive zero R,
and $Q(x)$ has exactly one real positive zero r.

Cauchy's theorem asserts that all zeros of $p(x)$ lie in the annular region

$$r \le |z| \le R.$$

How does one use these facts?

Think of them as heuristics, that give us a way of localizing the possible zeros of a polynomial. By localizing the zeros, we can guide the initial guesses of our numerical root finders.

Deflation

Once you have found a root r_1 of a polynomial consider next the deflated polynomial $q(x)$, where $q(x)$ satisfies

$$p(x) = (x - r_1)\, q(x).$$

One can obtain $q(x)$ by synthetic long-division:

$$x - r_1 \, \overset{\displaystyle q(x)}{\overline{\big)\, p(x)}}$$

To find a second root of $p(x)$, search for a root of $q(x)$, ETC.

Advantages:

- Avoid repeated convergence to the same root of $p(x)$.
- Reduce degree of relevant polynomial with each root (although: coefficient complexity may increase).

Caveats:

- Limited accuracy can build up errors in the roots as the polynomials are deflated.
- The order in which roots are found can affect the stability of the deflated coefficients (see p. 277 of NR:C).

Suggestions:

- If the synthetic division considers powers of x in decreasing order, then it is best to find the smallest root first, deflate, etc.
- Roots of deflated polynomials are really just "good suggestions". Use the original polynomial $p(x)$ to polish the roots.

Now let's look at some methods for finding roots of polynomials.

Newton's method applied to polynomials.

If $\quad p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$

then $\quad p'(x) = n a_n x^{n-1} + (n-1) a_{n-1} x^{n-2} + \cdots + a_1$

Recall the update rule for Newton's method: $\quad x_{m+1} = x_m - \dfrac{p(x_m)}{p'(x_m)}$

In order to evaluate $p(x)$ & $p'(x)$ we will make use of a small efficiency trick that is made possible by the nested Newton form for representing polynomials.

Earlier we gave a recursive procedure for evaluating $p(x)$, given its nested Newton form.
Here is an iterative procedure (with all centers equal to zero):

$$
\begin{aligned}
b_n &\leftarrow a_n \\
b_{n-1} &\leftarrow a_{n-1} + z b_n \\
&\vdots \\
b_i &\leftarrow a_i + z b_{i+1} \\
&\vdots \\
b_1 &\leftarrow a_1 + z b_2 \\
p(z) = b_0 &\leftarrow a_0 + z b_1
\end{aligned}
$$

This procedure evaluates the polynomial $p(x)$ at the point $x = z$, giv the polynomial coefficients $a_0, \ldots, a_n$.

Now observe that the intermediate quantities $b_0, \ldots, b_n$ computed by this iterative method actually serve another purpose as well. We can in fact write

$$p(x) = b_0 + (x-z)\,q(x)$$

with $$q(x) = b_n x^{n-1} + \cdots + b_2 x + b_1$$

Here x is a variable as usual, and z is the number that we used in the iterative evaluation of $p(z)$ to compute the $\{b_i\}$. So, in evaluating $p(z)$ as a number, we determine the coefficients $\{b_i\}$ of a polynomial $q(x)$ of degree $n-1$.

Here's the trick: When we differentiate we get

$$p'(x) = q(x) + (x-z)\,q'(x)$$

In particular,

$$p'(z) = q(z)$$

In other words, because $p(x)$ is a polynomial, we have a very simple method for computing its derivative. Indeed, when evaluating $p(z)$ we can simultaneously compute $p'(z)$.

Let us now write out Newton's method as applied to a polynomial.

Input:
- Initial guess x_0 of a root.
- The coefficients $a_0, \dots, a_n$ of the polynomial

Internal variables: x_m is the estimated root at the m^{th} step
$\{b_i\}$ used to evaluate $p(z)$ and define $q(x)$
$\{c_i\}$ used to evaluate $p'(z)$ (i.e., $q(z)$).

Pseudo-code:

```
For m = 0, 1, ... until termination conditions DO:

    z ← x_m
    b_n ← a_n
    c_n ← b_n

    For k = n-1 down to 1 DO:
        b_k ← a_k + z b_{k+1}
        c_k ← b_k + z c_{k+1}

    b_0 ← a_0 + z b_1

    x_{m+1} ← x_m - b_0 / c_1
```

(The code ignores error checks.)

Comments

Double roots

If $p(x)$ has a double root at $x=r$, then it is of the form

$$p(x) = (x-r)^2 h(x),$$

so

$$p'(x) = 2(x-r)\,h(x) + (x-r)^2 h'(x).$$

In other words, p & p' share a factor, and each of p & p' approaches zero as $x \to r$. Consequently, machine imprecision will dominate evaluation of the term $\dfrac{p(x)}{p'(x)}$ as $x \to r$.

Indeed, as $x \to r$ one divides one tiny number by one very small number.

As an additional consequence, to the extent that the method does converge, the rate of convergence is linear, not quadratic.

One can avoid the double-root problem by using numerical factors that seek quadratic factors directly (as opposed to linear factors, as with methods that seek single roots).

Seeking quadratic factors is also useful when looking for a pair of complex conjugate roots of a real polynomial. See NR:C, §9.5 for some details.

Of course this still leaves the difficulties associated with yet higher order roots. — To some extent one can detect the possibility of higher order roots, then use special techniques to find or rule-out such roots. Again, see NR:C, §9.5.

Deflation

The good news: Recall our previous representation, $p(x) = b_0 + (x-z)q(x)$, where $b_0, q(x)$ are computed incidentally during the evaluation of $p(z)$. If z is actually a root of $p(x)$, then $b_0 = 0$, so $q(x)$ is even the deflated polynomial $\frac{p(x)}{x-z}$!
In other words, the deflated polynomial is a byproduct of our Newton based root-finder.

The bad news: Polynomials can be very sensitive to variations in their coefficients. Consequently, after several deflations, the remaining roots may be very inaccurate.

Polishing

As we have said before, it is best to polish roots using a very accurate method, once one has found approximations to these roots. Certainly roots found from deflated polynomials should be regarded as approximate and perhaps worthy of polishing.
For double and higher-order roots, one may need to use special polishing techniques.

Newton's method is generally a good method for polishing both real and complex roots.

Global Methods

- We noted that Newton's method may fail to converge, if started "too far" from a root.
- Bisection will miss double roots.
- We haven't said how to find complex roots (although, starting Newton's method with a complex guess is certainly one way).

Are there globally convergent root-finders that can find all roots, whether real or complex or multiple?

The answer is yes. One such method is Laguerre's Method. This is discussed in NRiC. One possible drawback is that it requires complex arithmetic even when seeking real roots. Another method is Müller's Method, which we will now discuss. Its basic properties are:

- It often has global convergence. (Caveats apply)
- It can find any number of zeros, real or complex. It is even used on functions other than polynomials.
- It is an iterative method, with near quadratic convergence in the vicinity of a simple root.
- The basic approach is a generalization of the Secant method, but using quadratic interpolation instead of linear interpolation.

Müller's Method

We wish to find a zero of the function $f(x)$.

The inner loop of Müller's method operates as follows. Suppose our three prior estimates of a zero of $f(x)$ are the points x_0, x_1, x_2. To compute the next estimate we will construct the polynomial of degree two (or less) that interpolates $f(x)$ at x_0, x_1, x_2, then find one of its roots.

Recall that we can write the interpolating polynomial as

$$p(x) = f(x_2) + f[x_1, x_2](x - x_2) + f[x_0, x_1, x_2](x - x_2)(x - x_1).$$

Using the equality $(x-x_2)(x-x_1) = (x-x_2)^2 + (x-x_2)(x_2-x_1)$, we can rewrite $p(x)$ as

$$p(x) = f(x_2) + b(x - x_2) + a(x - x_2)^2,$$

where
$$a = f[x_0, x_1, x_2]$$

$$b = f[x_1, x_2] + f[x_0, x_1, x_2](x_2 - x_1)$$

If we then set $p(x)$ equal to zero and solve for x as the next approximation, we get

$$\boxed{x_3 = x_2 - \frac{2f(x_2)}{b \pm \sqrt{b^2 - 4af(x_2)}}}$$

ETC.

Comments:

- The roots of the quadratic equation ax^2+bx+c can be written either as $x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$ or as $x = \frac{-2c}{b \pm \sqrt{b^2-4ac}}$.

  For numerical stability, we chose the second form (recall that $c = f(x_2)$ which gets very small near a root of f).

- Again for numerical stability, it is a good idea to choose the $\pm$ sign in the denominator, $b \pm \sqrt{b^2-4ac}$, so as to make the magnitude of the denominator big.

- Because of the square-root in the expression for x_3, complex estimates are introduced automatically as needed.

- If f is a polynomial, then we are using another polynomial (of degree two or less) to estimate its roots. In particular, if f is itself a quadratic then of course we get its roots in one step.

Ex Let's return to our old example, with $f(x) = x^3 - x - 1$.
Here are the results of running Müller's method.

For each root we start the routine off with initial guesses $x_{-2} = 1$, $x_{-1} = 1.5$, and $x_0 = 2.0$. We terminate iteration when $|\Delta x_i| \le 5 \cdot 10^{-5}$.

First Root

i	x_i	$f(x_i)$
-2	1.0	-1
-1	1.5	.875
0	2.0	5
1	1.33333	.037037
2	1.32447	-.00105
3	1.324718	$1.44 \cdot 10^{-6}$
4	1.324718	$7.15 \cdot 10^{-13}$

The method estimates the first root to be $r_1 = 1.324718$.
The "accuracy" (caveat: this is really just the change in x_i) is $\sim 3.4 \cdot 10^{-7}$

Second Root Next we work with the deflated function $g(x) = \dfrac{f(x)}{x - r_1}$.

i	x_i	$g(x_i)$
⋮		
1	$-.6623697 + .5622605i$	$2.1355 \cdot 10^{-5} - 1.2 \cdot 10^{-5} i$
2	$-.66235898 + .5622795i$	$\sim 10^{-11}$

The method estimates the second root to be $r_2 = -.66236 + .56228i$ (with "accuracy" of $\sim 2.2 \cdot 10^{-5}$).

Third Root We can either take a complex conjugate to estimate the third root as $r_3 = -.66236 - .56228i$, or run Müller's method on the deflated function $h(x) = f(x)/(x-r_1)(x-r_2)$, which case we find the estimate $r_3 = -.66236 - .56228i$ (with an "accuracy" of $\sim 1.3 \cdot 10^{-5}$).

<u>Ex</u> Let's do another example in gory detail. We wish to find the roots of

$$p(x) = x^4 + 2x^2 - x - 1$$

We observed earlier that $p(x)$ has one positive root, one negative root, and one pair of complex conjugate roots.

Now let's look at the bound heuristics:

$$\rho = \min \left\{ n \frac{|a_0|}{|a_1|}, \sqrt[n]{\frac{|a_0|}{|a_n|}} \right\}$$

$$= \min \left\{ 4 \frac{1}{1}, \sqrt[4]{\frac{1}{1}} \right\}$$

$$= 1$$

So, there is at least one zero inside the circle of radius 1 about the origin.

$$r = 1 + \max_{0 \le k \le n-1} \left| \frac{a_k}{a_n} \right| = 1 + \max \{1, 1, 2, 0\}$$

$$= 3$$

So, all zeros of $p(x)$ lie inside the complex circle of radius 3 about the origin.

Conclusion: It makes sense to focus our search for zeros on the complex disk of radius 3.

We will start with the guess $x = 0$ (and, in order to get Müller going, the two points $x = \pm\frac{1}{2}$). We hope to find the zero with smallest magnitude, then deflate $p(x)$, find another zero, deflate further, and so forth.

The results of Müller's method follow.
(For each root we start the search with $x_{-2} = -\frac{1}{2}$, $x_{-1} = 0$, $x_0 = \frac{1}{2}$. We terminate the search when $|\Delta x_i| \le 5 \cdot 10^{-5}$.)

Root #1: After 5 iterations, Müller's method converges to $r_1 = .8251098$ with $|\Delta x_i| = \sim 2.4 \cdot 10^{-7}$.

Root #2: After 4 iterations, Müller's method converges to $r_2 = -.4818156$ with $|\Delta x_i| = \sim 1.4 \cdot 10^{-6}$.

Roots #3 & #4: After 2 iterations, Müller's method converges to $r_3 = -0.171647 + 1.576686i$ with $|\Delta x_i| = \sim 5.1 \cdot 10^{-6}$.
We then take $r_4 = -0.171647 - 1.576686i$.

Comments:

- In seeking r_2 we used the deflated function $\dfrac{f(x)}{x - r_1}$.
  In seeking r_3 we used the deflated function $\dfrac{f(x)}{(x-r_1)(x-r_2)}$.

- Given r_1, r_2, r_3, r_4 as above, we should next polish these estimates using the original function $f(x)$. In fact, it is easy to do so automatically within Müller's method. I used Newton's method to polish to an accuracy of 10^{-7}. This left the root estimates virtually unchanged.

- Notice that we didn't actually find the smallest root first.

- Notice also that two of the roots do lie outside the circle of radius 1, and that all lie inside the circle of radius 3 (about the origin).