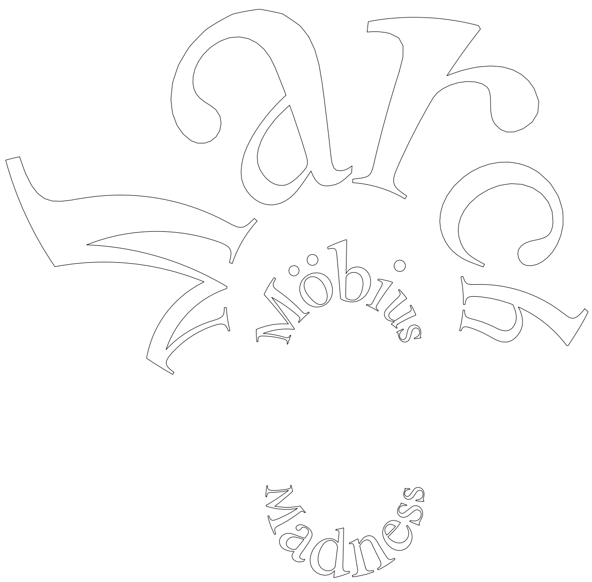
March Möbius Madness with a Polynomial PostScript March 32, 1995

Author: *Henry G. Baker, http://home.pipeline.com/˜hbaker1/home.html*; *hbaker1@pipeline.com*



Algorithms for extracting the roots of polynomials of the 2nd, 3rd and 4th degree are usually presented as cookbook recipes, with no insight about how such a recipe works or how it might have been discovered. This note shows how a clever high school student could use transformations (translation, scaling, rotation, inversion) of the complex plane to discover root-extraction methods for these polynomials.[1] In some instances, the transformations computed may be more valuable than the extracted roots. We are un-Abel to extend these methods to polynomials of higher degree.

This note also motivates the need for algebraic polynomial root-extraction by visualizing the chaos produced by a numerical Newton method. Code is included in the PostScript of this paper found in the directory `http://home.pipeline.com/˜hbaker1/sigplannotices/`.

[1] Garver [Garver29a] [Garver29b] independently made the same suggestion, as did Klein in 1888 [Klein56].

Digital Dentistry, or, Poly Wants a Cracker

Algorithms for polynomial root extraction have been known for about 500 years, but have on occasion been jealously guarded secrets.[2]

From the convoluted way these algorithms are presented today, one might conclude that they are still state secrets. The mathematics curriculum often seems to be geared primarily to the weeding out of non-mathematicians, rather than to providing mathematical insight for the rest of us. In this note, we attempt to provide some fun and insight along with some knowledge.

Excuse Newton's Dust, or, This Paper is Truly a 'Dusty Deck'

Many numerical analysis texts recommend the use of Newton's method to find the roots of a polynomial. Briefly, this method operates as follows. Given a polynomial $p(z)$, we make an initial guess z_0 at a root for the polynomial, and then compute a (hopefully improved) guess z_1 using the formula $z_1 = z_0 - p(z_0)/p'(z_0)$, where $p'(z)$ is the derivative of $p(z)$. This new guess z_1 is then used to compute an even better guess z_2 using the same formula, and so on. If z_0 is 'close enough' to an actual (non-repeated) root r, then Newton's method converges quadratically to the root, doubling the precision of the approximation with every iteration.

Newton's method for the square root of N is quite elegant: $z_{i+1} = (z_i + N/z_i)/2$—i.e., the improved guess is the arithmetic average (*mean*) of the current guess and N times the inverse of the current guess.[3]

[2] Tartaglia, an early solver of the cubic, met Ferrari, an early solver of the quartic, in a public math contest (and you thought TV *chess* was boring!) in Milan on August 10, 1548. Ferrari won and became a supervisor of tax assessments, a very lucrative job. Cardan, Ferrari's mentor, who "lived in great poverty until he became a lecturer in mathematics" (!), published solutions for both the cubic and the quartic in his *Ars Magna* (1545). [EB94]

[3] On the other hand, if we used the *geometric* average, then $z_{i+1} = \sqrt{z_i N/z_i} = \sqrt{N}$, and we would need only one step.

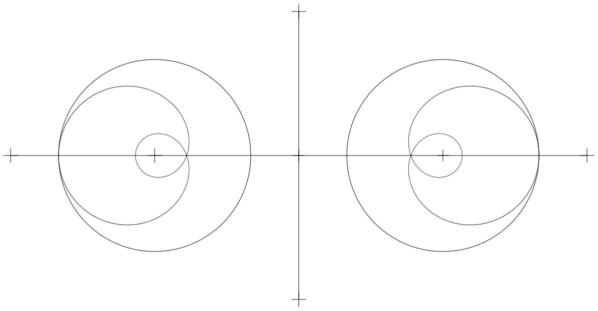


Figure 1: Images of circles $|z+1| = 2/3$ and $|z-1| = 2/3$ under Newton square root mapping $z \Rightarrow (z+1/z)/2$.

We can investigate the convergence properties of Newton's square root method in the complex plane by focussing on the case $N = 1$ — i.e., $z_{i+1} = (z_i + 1/z_i)/2$. This method is quite robust so long as z is not on the imaginary axis. Once the method starts on the left or right hand side of the imaginary axis, it stays there and converges to -1 or 1, respectively. However, if the method starts on the imaginary axis, it stays there forever, but jumps around quite a bit [Beeler72,#3(Schroeppel)]. We also notice that every point $w = (z + 1/z)/2$ has *two* preimage points: z and $1/z$ (because $z + 1/z = 1/z + 1/1/z$). Although computing the largest circle which guarantees convergence to 1 is problematic because both the center and the radius are unknown, it is easy to compute the largest circle centered on a root for which this method never escapes. We will call the radius of this circle the "Schwartzchild radius" for Newton's square root method, by analogy with the radius of a "black hole" in Einstein's general relativity. The "Schwartzchild" radius for the square root method is found by first noticing that if $0 < x < 1$, then $(x+1/x)/2 > 1$, and then solving the quadratic equation $((1-r)+1/(1-r))/2 = 1+r$, where $0 < r < 1$. This quadratic simplifies to $3r^2 - 2r = 0$ and has only 1 non-zero root — $r = 2/3$. Figure 1 is a plot of the 'black hole' surrounding each square root.

Newton's method for the cube root of 1 is similar to that for the square root of 1: $z_{i+1} = (2/3)z_i + (1/3)(1/z_i^2)$ — i.e., the weighted average of the current guess and the inverse square of the current guess.[4] Newton's cube root method is less robust than that for the square root. "Most" initial guesses still converge to a root,

[4]Note the lack of symmetry in this formula, compared with that for the square root—e.g., the 'weights' 2/3, 1/3 seem contrived. Cayley apparently agreed: "The solution is easy and elegant in the case of a quadratic equation, but the next succeeding case of the cubic equation appears to present considerable difficulty" [Cayley1879c].

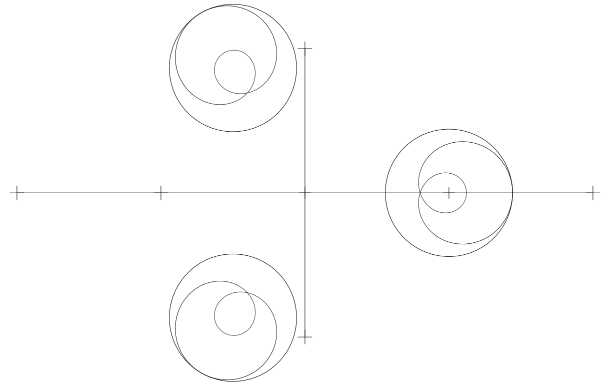


Figure 2: Images of circles around roots of radius 0.44174 under Newton cube root mapping $z \Rightarrow (2z + 1/z^2)/3$.

but the locations of the exceptions are no longer confined to one or more lines, but occupy regions which cover a substantial fraction of the complex plane. Since this method gives each point $w = (2/3)z + (1/3)(1/z^2)$ *three* preimage points (which can be found by solving a cubic equation), we can find the preimages of 0, the preimages of the preimages of 0, and so on. Thus, each exceptional point gives rise to three more, and thus in the n-th preimage generation, we have 3^n points—i.e., we have 3^n points which will be mapped to 0 in exactly n iterations of Newton's cube root method, and about $3^{n+1}/2$ points which map to 0 in n or fewer iterations. In Figure 3, we show a simple recursive PostScript program which explores n levels of this ternary tree and plots each one of these exceptional points to show the fractal nature of this set of exceptional points.

One might presume that with this fractal behavior, there are no 'nice' regions. This presumption is incorrect, as we can still compute a "Schwartzchild radius" for Newton's cube root method, by solving the cubic equation $(2/3)(1-r) + (1/3)(1-r)^{-2} = 1 + r$. This cubic simplifies to $5r^3 - 9r^2 + 3r = 0$, whose 2 non-zero roots are $(9 \pm \sqrt{21})/10$, the smaller of which is $(9 - \sqrt{21})/10 \approx 0.44174$. Once within a circle of this radius around a cube root of 1, Newton's method will never leave this circle again. Figure 2 is a plot of the 'black hole' surrounding each cube root. In Figure 3, we again plot these circles along with the myriad of exceptional points which hint at chaos.[5]

[5]Our plot of the preimages of 0 is essentially the same as Figure 43 of [Peitgen86], except that we also plot the Schwartzchild radius.

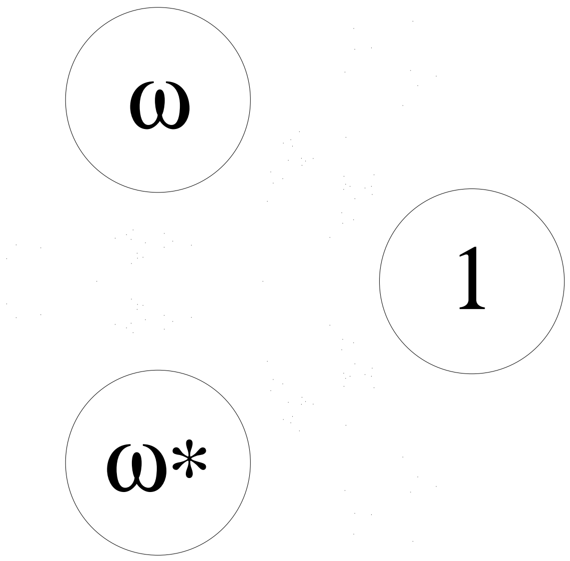


Figure 3: Preimages of 0 under the Newton cube root mapping $z \Rightarrow (2z + 1/z^2)/3$.

The existence of chaotic regions about these exceptional points shows how important it is to *start* Newton's method with a good initial guess. If we start within one of the 'black holes', then we are guaranteed to converge, and more importantly, we will converge monotonically (in terms of distance) to the closest root. We are therefore willing to perform some computations to ensure a good starting point for Newton's method. Unfortunately, it appears that for polynomials of degree 4 or less, computing a good initial point is as difficult as algebraically computing one of the roots. (Schur-fire techniques which test for the presence of zeros in a circular region are known, but they are outside the scope of this paper.)

Properties of Polynomials

The n-degree polynomial $p(z) = a_n z^n + a_{n-1} z^{n-1} + ... + a_1 z + a_0$ has exactly n complex *roots* r_i, such that $p(r_i) = 0$. In particular, $p(z)$ can be factored into a product of *linear factors* $z - r_i$, together with the high-order coefficient a_n.

$$p(z) = a_n \prod_i (z - r_i)$$

Two facts are immediately evident from this representation: the *sum* of the roots can be calculated from a_{n-1}, and the *product* of the roots can be calculated from a_0:

$$\sum_i r_i = -a_{n-1}/a_n \qquad \prod_i r_i = a_0/a_n$$

The *center of mass* of the roots is $-a_{n-1}/na_0$ — the root sum divided by n. Thus, we can easily compute $p(z - a_{n-1}/na_o)$, which is $p(z)$ translated to a coordinate system in which the center of mass is now the origin, and the coefficient of z^{n-1} in $p(z - a_{n-1}/na_0)$ is thus zero.

If $p(z)$ is divided synthetically by $z - c$, we get a quotient $q(z)$ and a remainder of degree smaller than 1 — i.e., a constant. Thus, $p(z) = q(z)(z - c) + d$. But $p(c) = q(c)(c - c) + d = d$. Thus, the remainder of $p(z)$ divided by $z - c$ is $p(c)$.

If one of the roots r_i is *zero*, then the corresponding linear factor $z - r_i = z - 0 = z$, in which case the polynomial is a multiple of z and its constant coefficient is zero. Thus, a zero root is obvious from the constant coefficient of zero.

If $r_i = r_j$, for some $i \neq j$, then we have a *repeated* root, where the *multiplicity* of the root is the number of occurrences of the root in the factored polynomial. Using calculus, if we compute the derivative $p'(z)$ of the polynomial $p(z)$, then the multiplicity of a repeated root in the derivative polynomial is one less than its multiplicity in the original polynomial $p(z)$. In other words, if $p(z) = q(z)(z - r)^m$, then $p'(z) = q'(z)(z - r)^m + q(z)m(z - r)^{m-1} = (z - r)^{m-1}(q'(z)(z - r) + mq(z))$, assuming that r is not a root of $q(z)$.

Using this fact, we can easily find all repeated roots by computing the greatest common divisor of a polynomial $p(z)$ and its derivative $p'(z)$ using polynomial "synthetic division".

$$\begin{aligned}
&a_n(z - r)^{m-1} \\
&= \gcd(p(z), p'(z)) \\
&= \gcd(q(z)(z - r)^m, (z - r)^{m-1}(q'(z)(z - r) + mq(z))) \\
&= (z - 1)^{m-1} \gcd(q(z)(z - r), q'(z)(z - r) + mq(z))
\end{aligned}$$

Abel was I ere I saw Elba

Somewhat less well known is the fact that if r_i is a root of the polynomial $p(z)$, then $1/r_i$ is a root of the *reversed* polynomial $r(z)$ in which the coefficients are listed *back-*

wards.[6]

$$r(z) = a_0 z^n + a_1 z^{n-1} + ... + a_{n-1} z + a_n$$
$$= z^n p(1/z)$$

We find this fact useful when the reversed polynomial has more desirable properties than the original polynomial.

An interesting case is that of a *polynomial palindrome* — a polynomial which is the same, both forwards and backwards (also called a *reciprocal* polynomial [Lovitt39] [Eves66]).[7]

The degree of palindromic polynomials can be reduced, using the fact that *the inverse* $1/r$ *of every root* r *is also a root*. If a palindromic polynomial is of *odd* degree, then -1 must be a root because it must be its own reciprocal, and the linear factor $z+1$ can then be divided out using synthetic polynomial division. We therefore consider a palindromic polynomial of *even* degree. By dividing the $2n$-degree palindromic polynomial $p(z)$ by z^n and pairing z^i with z^{-i}, we get

$$p(z)/z^n = 2a_0(z^n + 1/z^n) + 2a_1(z_{n-1} + 1/z^{n-1}) + ... + a_n$$

Consider the functions defined recursively by $V_0(z) = 2$, $V_1(z) + z + 1/z$, $V_p(z) = (z+1/z)V_{p-1}(z) - V_{p-2}(z)$. By induction,

$$V_p(z) = (z + 1/z)V_{p-1}(z) - V_{p-2}(z)$$
$$= z^p + 1/z^p$$

So we can now represent $p(z)/z^n = 2a_0 V_n(z) + 2a_1 V_n - 1(z) + ... + a_0$. But $V_p(z)$ is a degree-p polynomial in $z' = z + 1/z$, so $p(z)/z^n = q(z + 1/z)$, where $q(z)$ is only *half* the degree of $p(z)$.

Möbius Transforms

The Möbius transform is one of the most fascinating transforms of the complex plane. Given 4 complex numbers A, B, C, D, the Möbius transform with these conformal (!) parameters maps the complex number z into the number $(Az+B)/(Cz+D)$. So long as $AD - BC \neq 0$, this mapping is invertible. The Möbius transform is the algebraic closure of the transforms generated by translation ($z \Rightarrow z + B$), dilation ($z \Rightarrow az$), rotation ($z \Rightarrow e^{i\theta}$), and inversion ($z \Rightarrow 1/z$).

[6] The characteristic polynomial of the matrix A^{-1} is a scalar multiple of the reverse of that of A.

[7] The characteristic polynomial of an *orthogonal* matrix A ($AA^T = I$) is a reciprocal polynomial.

You may have already seen a Möbius transform before without realizing it. For example, the complex *arctangent* function is a rotated and scaled version of a Möbius-transformed complex *logarithm* function:

$$\arctan(z) = \frac{i}{2} \log\left(\frac{1 - iz}{1 + iz}\right)$$

Möbius transforms are best understood as linear transformations on complex numbers expressed in homogeneous coordinates—i.e., pairs (x, y) of complex numbers representing the ratio $z = x/y$. Then the Möbius transform $(Az + B)/(Cz + D)$ can be represented in 2×2 matrix form as:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}\begin{pmatrix} z \\ 1 \end{pmatrix} = \begin{pmatrix} Az + B \\ Cz + D \end{pmatrix}$$

This form is more than suggestive—the composition of two Möbius transforms is represented by matrix multiplication of the associated matrices. Furthermore, the inverse of a given Möbius transform is represented by the inverse of the matrix of the given transform. However, due to the homogeneous nature of the representation, any scalar multiple of a Möbius transform matrix represents the same transform. We can use this property to "clear fractions" from the representation of the inverse transform of $(Az + B)/(Cz + D)$:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}\begin{pmatrix} D & -B \\ -C & A \end{pmatrix} = \begin{pmatrix} D & -B \\ -C & A \end{pmatrix}\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$
$$= \begin{pmatrix} AD - BC & 0 \\ 0 & AD - BC \end{pmatrix}$$

Some simple properties of Möbius transforms are: 1) in the identity transform, $A = D \neq 0$ and $B = C = 0$; 2) in a translation, $A = D \neq 0$, $C = 0$; 3) in a scaling and/or rotation, $A \neq 0$, $D \neq 0$, $B = C = 0$; 4) in a scaling/rotation plus a translation (a *similarity* transformation), $A \neq 0$, $D \neq 0$, $C = 0$; and 5) in an inversion, $B = C \neq 0$, $A = D = 0$. The transform in which $A \neq 0$, $C \neq 0$, $D \neq 0$, $B = 0$ corresponds to an inversion, followed by a similarity transformation, followed by another inversion.

The 3 distinct images of 3 distinct points determine a unique Möbius transform. A simple way to construct this transform is the following. The Möbius transform mapping $a \Rightarrow 0$, $b \Rightarrow \infty$, $c \Rightarrow 1$ can easily be seen to be

$$z \Rightarrow \left(\frac{c - b}{c - a}\right)\left(\frac{z - a}{z - b}\right),$$

which has the associated matrix

$$\begin{pmatrix} c-b & 0 \\ 0 & c-a \end{pmatrix}\begin{pmatrix} 1 & -a \\ a & -b \end{pmatrix}$$

The Möbius transform mapping $a \Rightarrow a'$, $b \Rightarrow b'$, $c \Rightarrow c'$ can then be constructed as the composition of the mappings $a \Rightarrow 0 \Rightarrow a'$, $b \Rightarrow \infty \Rightarrow b'$, $c \Rightarrow 1 \Rightarrow c'$, which can be computed by multiplying the matrix of the type above associated with the first mapping by the matrix (which is the inverse of a matrix of the type above) associated with the second mapping. For example, we can construct a Möbius mapping which permutes the 3 cube roots of 1 by exchanging 1 and $(-1+\sqrt{3})/2$, but leaving $(-1-\sqrt{3})/2$ fixed: $z \Rightarrow (-1+\sqrt{3})/2z$.

Three distinct points in the plane determine a circle or a line (a line is a "circle through infinity"). Given any other 3 distinct points, one may construct a Möbius transform. This Möbius transform maps not only the 3 given points to their images, but it also maps *every* point on the circle determined by the first 3 points into points on the circle determined by the second 3 points. We thus have the phrase "*Möbius transforms map circles into circles*". This property means that the class of graphical objects constructed solely of *points*, *lines* and *circular arcs* is closed under Möbius transforms, because any such object is transformed into another object of the same class.[8]

We will need some derivatives of the Möbius transform.

$$m(z) = \frac{Az+B}{Cz+D}$$

$$m'(z) = \frac{AD-BC}{(Cz+D)^2} = \frac{\Delta}{(Cz+D)^2},$$

where

$$\Delta = \begin{vmatrix} A & B \\ C & D \end{vmatrix}$$

$$m''(z) = \frac{-2C\Delta}{(Cz+D)^3}$$

$$m'''(z) = \frac{6C^2\Delta}{(Cz+D)^4}$$

Stereographic Projection

Closely related to the Möbius transformation of the complex plane is the *stereographic projection* which maps the surface of a sphere onto the complex plane in an invertible manner. Briefly, a stereographic projection can be achieved by placing a sphere—e.g., a globe—on top of the complex plane in such a way that the south pole of the sphere sits on the origin of the complex plane. The projection is then obtained by considering a ray cast from the north pole of the globe through any other point on the globe and then onto the plane. The point (other than the north pole) intersected by the ray on the sphere is then mapped via the projection onto the point intersected by the ray on the plane. Furthermore, this mapping is invertible if we associate the north pole of the sphere with ∞.

It is not essential that the sphere sit on top of the plane, and it is more convenient if the plane cuts the sphere at its equator. Thus, if the sphere has radius 1, then we can compute the point $a+bi$ that the point (x,y,z) on the sphere maps into and vice versa.

$$(x,y,z) \Rightarrow \frac{x+yi}{1+z}$$

and

$$a+bi \Rightarrow \left(\frac{2a}{1+a^2+b^2}, \frac{2b}{1+a^2+b^2}, \frac{1-a^2-b^2}{1+a^2+b^2}\right)$$

The stereographic projection is particularly elegant because it makes more explicit the symmetry of the mapping $a+bi \Rightarrow 1/(a+bi)$ as well as the symmetry of the other Möbius mappings. For example, the mapping $a+bi \Rightarrow -a-bi$ induces the mapping $(x,y,z) \Rightarrow (-x,-y,z)$ on the globe, which is simply a rotation by $180°$ about the N-S pole. Similarly, the mapping $a+bi \Rightarrow 1/(a+bi)$ induces the mapping $(x,y,z) \Rightarrow (x,-y,-z)$ on the globe, which is simply a rotation by $180°$ about the pole through the equator at $0° - 180°$ (the "a" axis). The mapping $a+bi \Rightarrow -1/(a+bi)$ induces the composition of the previous two rotations to induce the mapping $(x,y,z) \Rightarrow (-x,y,-z)$ on the globe, which is a rotation by $180°$ about the pole through the equator at $90° - 270°$ (the "b" axis).

The most elegant part of the stereographic projection is the fact that circles on the globe are mapped into circles and lines on the complex plane, and conversely, lines and circles on the complex plane are mapped into circles on the globe. Thus, every Möbius transformation of the complex plane induces a transformation of the globe in such a way that circles on the globe are transformed into other circles on the globe.

Not all Möbius transformations induce merely rotations of the globe. For example, Möbius transforms can induce mappings of great circles (such as the equator)

[8] Simple "polylines" in the *AutoCAD(r)* drawing system from Autodesk, Inc., consist of lines and circular arcs.

into smaller circles (such as a latitude), as well as inducing mappings of 3 equidistant points on the globe into 3 non-equidistant points. However, those Möbius transformations which *do* induce rotations can be expressed with just two complex parameters: $(Az+B)/(-B^*z+A^*)$, where A^*, B^* are the conjugates of A, B, respectively. It is not mere coincidence, then, that we find that this Möbius subgroup is isomorphic to the (unit) *quaternions*, which provide an elegant way to express arbitrary rigid rotations in three dimensions.

The stereographic projection can be used to map the roots of a polynomial onto a globe in such a way that the symmetry between a polynomial and its reverse polynomial is evident. Since reversing a polynomial transforms each root r to its inverse $1/r$, and since the mapping $z \Rightarrow 1/z$ is simply a $180°$ rotation of the globe, the original roots plotted on the globe will simply follow the globe during its $180°$ rotation, while the arrangement of the roots is otherwise unchanged. The stereographic projection also makes the complex *Fourier Transform* more elegant—the Fourier Transform of a polynomial is the evaluation of the polynomial at evenly spaced intervals around the equator of the stereographic globe.

Visualization Using PostScript(tm)

Although the *PostScript* language [Adobe90] was developed primarily as a *page description language* for use in modern computer printers, this Forth-like language is surprisingly powerful. We will use PostScript as both an "application" language for symbolic and numeric computing, in addition to using it as a graphical language for visualizing the results of these computations. In so doing, we will be doing the ultimate in "lazy evaluation", since the entire computation will be performed inside the printer in a "procrastinating PostScript"!

PostScript is a stack-oriented language—all arguments to a function are provided on the stack, and all results from the function are returned on the stack. Thus, the PostScript `add` function takes the two numbers from the top of the stack and replaces them with a single number—the sum of these two arguments. PostScript also provides literal numbers and interactive execution (even on your laser printer![9]), so typing the following will compute 2+3:

```
>2 3 add ==
5
```

(The `==` command/function prints the top element of the stack and pops it off; `pstack` prints the whole stack without popping it; `clear` clears the stack.) Other PostScript arithmetic operators include `neg`, `sub`, `abs`, `mul`, `div`, `mod`, `sin`, `cos`, `exp`, `ln`, `sqrt`, etc.

We will not be long content with the built-in PostScript operators, so we will want to construct our own. Two operations that are often required in PostScript are `add1` and `sub1`, which increment and decrement the number at the top of the stack, respectively.

```
>/add1 {1 add} def
>/sub1 {1 sub} def
>3 add1 ==
4
>3 sub1 ==
2
```

(The "`/`" quotes the name "`add1`" so that this symbol is treated literally, rather than having the interpreter try to execute it.)

Now suppose that we want to define a general "`add-n`" function. Rather than having to define each of the functions `add2`, `add3`, etc., separately, we can define a generic function-producing function `add-n` which takes an argument specifying the increment, and then returns the appropriate *function* on the stack.

```
>/add-n {{add} 1 make-closure} def
>/add2 2 add-n def
>/add3 3 add-n def
>3 add2 ==
5
```

Before looking at the non-primitive operator `make-closure`, let us first look at the code that `add-n` constructs.

```
>2 add-n ==
{2 add}
```

So `add-n` is working correctly. But what does `make-closure` do?

```
>/make-closure
 {exch aload length dup 2 add
  -1 roll add array astore cvx}
 def
```

[9] Hook up a serial line from your computer to the PostScript printer; start up your terminal emulation program in 7-bit ASCII mode; and type `executive<cr>`. Although the characters '`executive`' will *not* be echoed, your efforts should be rewarded with a '>' PostScript interactive prompt character [Adobe90,s2.4.4].

Without going into detail, `make-closure` simply creates a new operator by appending some number of literals from the top of the stack, thus incorporating these parameters directly into the code.

We will require some number of operations on vectors of elements, constructed in PostScript using square brackets '`[]`', for which PostScript provides a little native support. Let us first make a vector of length n of integers from 0 up to n, which operation is known as `iota` from the APL language [Brown88]:

```
>/iota
 {0 1 3 -1 roll {} for array astore}
 def
>5 iota ==
[0 1 2 3 4]
```

We would like to *map* a function over a vector to produce a vector of results; `map1` maps a function of 1 argument:

```
>/map1 {[ 3 1 roll forall ]} def
>[3 1 4 1 5] {add1} map1 ==
[4 2 5 2 6]
```

We would also like to map a function of 2 arguments over 2 vectors to produce a vector of results. Such a function can sum 2 vectors together element-by-element, and we call it `map2`:

```
>/map2
 {exch dup dup length array copy
  dup length 1 sub 0 1 3 -1 roll
  {4 index 1 index get 3 index
   2 index get 5 index exec
   2 index 3 1 roll put} for
  4 1 roll pop pop pop}
 def
>[3 1 4 1 5 1] [2 7 1 8 2 8]
 {add} map2 ==
[5 8 5 9 7 9]
```

Using these functions, we can now program *complex numbers* into PostScript, using the representation `[3 4]` for $3 + 4i$. In this paper,[10] we give definitions of *complex* number functions, where a "`c`" is prepended to the name of each PostScript *real* function—e.g., `cadd`, `conjugate`, `norm`, `cscale`, `cdiv`:

```
>/cadd {{add} map2} def
>/conjugate
 {[ exch aload pop neg ]}
 def
>/cnorm
 {dup conjugate cmul aload pop pop}
 def
>/cscale
 {{mul} 1 make-closure map1}
 def
>/cdiv
 {conjugate dup cnorm 1 exch div
  cscale cmul}
 def
```

We are now in a position to program polynomials with complex coefficients. We represent the polynomial $p(z)$

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + ... + a_1 z + a_0$$

having complex coefficients a_i by the "little-endian" (reversed) PostScript vector

$$[a_0 a_1 ... a_{n-1} a_n]$$

For example, the polynomial $x^2 - ix + 3$ is represented by the vector `[[3 0] [0 -1] [1 0]]`.

We will require several operations on polynomials, including polynomial evaluation at a particular point, polynomial differentiation, etc. For example, a function $n(z)$ to compute an improved Newton approximation to a root r is given by $n(r) = r - p(r)/p'(r)$, where $p(z)$ is a polynomial and $p'(z)$ is its derivative, can be computed by `pc-newton`:

```
>/pc-newton % make a procedure
            % to do a newton
            % iteration on
            % a polynomial p.
 {dup pdiff exch
  {2 index peval exch
   2 index peval cdiv csub}
  2 make-closure}
 def
>[2 2] [[-1 0] [0 0] [1 0]]
 pc-newton exec ==
[1.125 0.875]
```

Solving a linear polynomial

We solve a polynomial in an excruciatingly pedantic way, in order to give insight about how to deal with higher-order polynomials. The solution of the linear polynomial

[10] "In this paper" is meant quite literally—if you have the PostScript code for this paper, you have the program code!

$a_1z + a_0 = 0$ proceeds in two steps. First, we make the polynomial *monic*[11] by dividing by a_1 to get $p(z) = z + a_0/a_1 = 0$. We know that a polynomial of degree 1 has exactly one root, and this root can be transformed by a simple *translation* into a canonic location—e.g., the origin.

Consider the substitution $z = y + B$. Then $p(z) = p(y+B) = (y+B)+a_0/a_1 = y+(B+a_0/a_1) = q(y)$. The polynomial $q(y)$ will be in the canonic form $y - 0$ iff $B + a_0/a_1 = 0$. Solving for B, we get $B = -a_0/a_1$. We can now calculate the root by substituting $y = 0$ into the equation $z = y + B = 0 + B = -a_0/a_1$.

Solving a quadratic polynomial

The traditional high-school derivation of the solution of the general quadratic equation involves "completing the square". I never understood the motivation for this "stupid root trick" when I was in high school, and I still don't understand it today. My best motivation comes from an understanding that a quadratic equation has exactly 2 roots, and so long as the roots are distinct, there exists a simple *similarity transformation* (scaling/rotation plus translation) that will map these 2 roots onto the two points ± 1 — i.e., we want to transform any given quadratic equation having distinct roots into the canonic form $z^2 - 1 = 0$, whose solution we know because we constructed this equation as $z^2 - 1 = (z-1)(z+1)$. The inverse of the similarity transformation applied to ± 1 will then yield the roots of the original equation.

First, make the quadratic polynomial monic by dividing by its high order coefficient. We then have $p(z) = z^2 + a_1z + a_0$. Consider the substitution $z = Ay + B$. Using *Taylor series*, extract the coefficients of the substituted polynomial:

$$\begin{aligned} p(z) &= p(Ay + B) \\ &= y^2A^2p''(B)/2 + yAp'(B) + p(B) \\ &= A^2y^2 + A(2B + a_1)y + B^2 + a_1B + a_0 \\ &= A^2q(y), \end{aligned}$$

$q(y)$ is a monic quadratic.

If we equate the coefficients of this equation to that of the canonic equation $y^2 - 1$, then we have 2 equations in the two unknowns A, B:

$$Ap'(B)/(A^2p''(B)/2) = p'(B)/A = 0,$$

[11] Since a monic polynomial has one less degree of freedom, making a polynomial monic is "monic depression".

or more simply, $p'(B) = 2B + a_1 = 0$.

$$p(B)/(A^2p''(B)/2) = p(B)/A^2 = -1,$$

or more simply, $A^2 = -p(B)$.

The first equation is trivially solved for $B = -a_1/2$, whereupon the second equation becomes $A^2 = -p(-a_1/2)$. This equation can be solved by extracting roots: $A = \pm\sqrt{-p(-a_1/2)}$. We can now recover the roots of the original equation by substituting ± 1 into the equation $z = Ay + B$:

$$\begin{aligned} r_1 &= A(1) + B \\ &= \sqrt{-p(B)} + B \\ &= \sqrt{-p(-a_1/2)} - a_1/2 \\ &= -a_1/2 + \sqrt{(-a_1/2)^2 - a_0} \end{aligned}$$

$$\begin{aligned} r_2 &= A(-1) + B \\ &= -\sqrt{-p(B)} + B \\ &= -\sqrt{-p(-a_1/2)} - a_1/2 \\ &= -a_1/2 - \sqrt{(-a_1/2)^2 - a_0} \end{aligned}$$

We now recognize the traditional quadratic equation formulae. We first note that *both* solutions for A work—one maps the roots r_1, r_2 onto 1,-1, respectively, while the other maps the roots r_1, r_2 onto -1,1, respectively. We also note that although we assumed that the roots were distinct, the final formulae work even when the roots are the same! In this case, the rotation/scaling factor $A = 0$, so that the two roots ± 1 map into the one root of $p(z)$.

Solving a cubic polynomial

If we translate a cubic polynomial to the center of mass of its roots, we produce a cubic polynomial whose quadratic coefficient is zero. If after this translation the linear coefficient—which can be Taylored as $p'(-a_2/3a_3)$ — is also zero, then we have a trivial cubic equation of the form $b_3(z + a_2/3a_3)^3 + b_0 = a_3(z + a_2/3a_3)^3 + p(-a_2/3a_3) = 0$, which is easily solved by extracting the 3 cube roots of $-b_0/b_3 = -p(-a_2/3a_3)/a_3$. Thus, this trivial case can be solved with a simple *similarity* transformation

$$\begin{aligned} z &= Ay + B \\ &= y\sqrt{-b_0/b_3} + B \\ &= y\sqrt{-p(B)/a_3} + B \\ &= y\sqrt{-p(-a_2/3a_3)/a_3} - a_2/a_3 \end{aligned}$$

We can recover the 3 roots of $p(z)$ by substituting 1, $(-1+\sqrt{3})/2$, and $(-1-\sqrt{3})/2$ for y in the above equation.

This trivial case having been disposed of, we can now focus on the hard case of the cubic—where the linear coefficient is *not* zero after transforming to the center of mass coordinate system. We know from our study of Möbius transformations that they can transform lines and circles into lines and circles. In particular, they can transform any circle (or line) defined by 3 points into any other circle (or line) defined by the transform of each of these 3 points. Therefore, we suspect that a Möbius transform could be used to transform a cubic equation $p(z) = z^3 + bz + 2 + cz + d = 0$ having distinct roots into the canonic equation $z^3 - 1 = 0$ in such a way that the distinct roots of $p(z)$ are mapped onto the 3 cube roots of one — 1, $(-1+\sqrt{3})/2$, $(-1-\sqrt{3})/2$. A possible problem is that the Möbius parameters might not be easily calculated from the coefficients of the given cubic equation.

We now consider the Möbius transformation $z = (Ay + B)/(Cy + D)$. If this transformation is nonsingular, then we cannot have $C = D = 0$. Since the trivial similarity case has already been disposed of, we must have $C \neq 0$. For if $C = 0$, then the simple similarity transformation $(A/D)y + (B/D)$ could solve $p(z)$, but we already know that it can't. Therefore, $C \neq 0$. Less obviously, we must also have $D \neq 0$. Consider the case $D = 0$. Then

$$\begin{aligned} &p((Ay+B)/(Cy+D))(Cy+D)^3 \\ &= p((Ay+B)/Cy)(Cy)^3 \\ &= B^3 p((A/C) + 1/(Cy/B))(Cy/B)^3 \\ &= B^3 p((A/C) + 1/x)x^3 \end{aligned}$$

But this is a polynomial translated by A/C, and then reversed. If the two middle coefficients are both zero, then reversing the polynomial will not change that, so the translation by A/C alone must have made them both zero. But this is the same trivial case that we have already disposed of above.

If $z = (Ay + B)/(Cy + D)$ in the equation $p(z)$, we have $Eq(y) = p(z)(Cy + D)^3 = p((Ay + B)/(Cy + D))(Cy + D)^3$, where $q(y)$ is a monic polynomial, and where E is a function of A, B, C, D, and the coefficients of $p(z)$; E must now be determined. The coefficient of y^3 in the polynomial $Eq(y)$—i.e., E itself—can be found via Taylor series as $Eq'''(0)/3!$, but the algebra becomes quite involved. We therefore consider Taylor series applied to the reverse/backwards polynomial $Er(y)$ associated with $Eq(y)$:

$$\begin{aligned} Er(y) &= Ey^3 q(1/y) \\ &= y^3 p((A/y + B)/(C/y + D))(C/y + D)^3 \\ &= p((A + By)/(C + Dy))(C + Dy)^3. \end{aligned}$$

Therefore, $E = Er(0) = p(A/C)C^3$. The coefficient of y^2 in $q(y)$ can also be computed by Tayloring $r(y)$. By setting the 3 low order coefficients of $q(y)$ to be the same as $y^3 - 1$, we get 3 equations:

$$\begin{aligned} q''(0)/2! &= r'(0) \\ &= (D/C)\frac{3p(A/C) + (B/D - A/C)p'(A/C)}{p(A/C)}, \end{aligned}$$

or simply $3p(A/C) - (A/C - B/D)p'(A/C) = 0$.

$$q'(0) = (D/C)^2 \frac{3p(B/D) + (A/C - B/D)p'(B/D)}{p(A/C)},$$

or simply $3p(B/D) + (A/C - B/D)p'(B/D) = 0$.

$$q(0) = (D/C)^3 \frac{p(B/D)}{p(A/C)} = -1,$$

or more simply,

$$(D/C)^3 = -\frac{p(A/C)}{p(B/D)}.$$

Since A/C and B/D figure so prominently in our equations, we substitute $r = A/C$ and $s = B/D$:

$$\begin{aligned} 3p(r) - (r - s)p'(r) &= 0 \\ 3p(s) + (r - s)p'(s) &= 0 \\ (D/C)^3 &= -\frac{p(r)}{p(s)} \end{aligned}$$

It should be obvious that subtracting the first from the second equation gives an equation which is divisible by $r - s$:

$$(r - s)(3rs + a_2(r + s) + a_1) = 0,$$

or more simply,

$$3rs + a_2(r + s) + a_1 = 0$$

If we subtract s times the first equation from r times the second equation, we get

$$(r - s)(a_2 rs + a_1(r + s) + 3a_0),$$

or more simply,

$$a_2 rs + a_1(r+s) + 3a_0 = 0$$

Since we have 2 linear equations in the 2 unknowns $r+s$ and rs, we can solve for them. Furthermore, given $r+s$ and rs, we can construct the quadratic equation $x^2-(r+s)x+rs=0$, which can be solved (see above) for r and s. Thus, our constructed quadratic equation for r and s (after clearing fractions) is

$$(3a_1 - a_2^2)x^2 + (9a_0 - a_1a_2)x + 3a_0a_2 - a_1^2 = 0$$

Now that we have r, s in hand, we can solve for D/C using the third equation:

$$D/C = \sqrt[3]{-p(r)/p(s)}$$

If we now arbitrarily set $C = 1$, then we have all 4 Möbius parameters:

$$\begin{aligned}
A &= C(A/C) = r \\
B &= D(B/D) = s\sqrt[3]{-p(r)/p(s)} \\
C &= 1 \\
D &= D/C = \sqrt[3]{-p(r)/p(s)}
\end{aligned}$$

Using these parameters, we can Möbius transform the 3 cube roots of 1 into the roots of $p(z)$:

$$z = \frac{Ay+B}{Cy+D} = \frac{ry + s\sqrt[3]{-p(r)/p(s)}}{y + \sqrt[3]{-p(r)/p(s)}},$$

$y = 1$, $(-1+\sqrt{3})/2$, $(-1-\sqrt{3})/2$.

We can simplify our constructed equation as:

$$rs(x) = x^2 + \frac{3p\left(\frac{-a_2}{3a_3}\right)}{p'\left(\frac{-a_2}{3a_3}\right)}x - \frac{p'\left(\frac{-a_2}{3a_3}\right)}{3} = 0.$$

This equation is equivalent to the traditional cubic "discriminant" equation $u^2 - p(-a_2/3a_3)u - (p'(-a_2/3a_3)/3)^3 = 0$ with the substitution $x = -3u/p'(-a_2/3a_3)$.

We were sloppy in our derivation, because we assumed that $r - s \neq 0$. However, if $r = s$, then our Möbius transform is singular, which cannot happen if the roots of $p(z)$ are distinct. Furthermore, we assumed that $p(r) \neq 0$ and $p(s) \neq 0$. But $p(r) = 0$ or $p(s) = 0$ only if the degree of $\gcd(p(z), rs(z))$ is greater than zero. When this gcd computation is performed symbolically, we find that $p(z)$ and $rs(z)$ can share roots only when the roots of $p(z)$ are not distinct or $p'(-a_2/3a_3) = 0$. But $p'(-a_2/3a_3) = 0$ implies the trivial case which is handled by a similarity transform. Therefore $p(r) \neq 0$ and $p(s) \neq 0$.

We can therefore compute the roots of $p(z)$ using matrix multiplication, as follows:

$$\begin{pmatrix} 3 & -a_2 \\ 0 & 3 \end{pmatrix}\begin{pmatrix} r & s\sqrt[3]{\frac{-p(r)}{p(s)}} \\ 1 & \sqrt[3]{\frac{-p(r)}{p(s)}} \end{pmatrix}\begin{pmatrix} 1 & \frac{-1+\sqrt{3}}{2} & \frac{-1-\sqrt{3}}{2} \\ 1 & 1 & 1 \end{pmatrix}$$

except that if both inner coefficients are zero after translating to the center of mass, replace the middle matrix with

$$\begin{pmatrix} \sqrt[3]{-a_0} & 0 \\ 0 & 1 \end{pmatrix}$$

We thus have an algorithm for a cubic polynomial. First, compute $g(z) = \gcd(p(z), p'(z))$; $g(z)$ will be nontrivial only when there are repeated roots. If the degree of $g(z)$ is 2, then we have a triple root, and $p(z) = (z-r)^3$, so $r = -a_2/3$. If the degree of $g(z)$ is 1, then we have a double root, $g(z)$ is trivially solved to produce that root, and the third root is obtained from $p(z)$ by synthetic division. If the degree of $g(z)$ is 0, then we have 3 distinct roots, and can then construct a Möbius transform to map these roots into the cube roots of 1. Shift the polynomial to the center of mass by using the substitution $w = z + a_2/3$. If now $a_1 = a_2 = 0$, then we have the 3 roots that are the cube roots of $-a_0$. Finally, construct the Möbius transform according to the above equations, and then use it to map the cube roots of 1 into the roots of the given equation.

Although a Möbius mapping can transform any cubic having distinct roots into the form $z^3 - 1 = 0$, this mapping is not unique because there are 3! = 6 permutations of the cube roots of 1, and any permutation of these cube roots can be Möbius-mapped into the roots of the given cubic. Thus, if we have a Möbius mapping from the cube roots of 1 into the roots of the given cubic, we can always compose it with a Möbius mapping which permutes the cube roots of 1 to produce a different Möbius mapping of the cube roots of 1 into the roots of the given cubic.

Solving the general cubic equation essentially "uses up" all the degrees of freedom of the Möbius transform. Given any (non-singular) Möbius transform $(Az+B)/(Cz+D)$, we can transform the cube roots of 1 into their images r_1, r_2, r_3 under the transform. So long as $r_i \neq \infty$, the product $(z-r_1)(z-r_2)(z-r_3)$ is a cubic polynomial. If one of the r_i is ∞, say $r_3 = \infty$, then our cubic is "really" a quadratic with 2 distinct finite roots. Of course, for any cubic polynomial obtained as the image of a Möbius transform, there are 5 other Möbius transforms which map into the same polynomial.

We note that transforming a general cubic equation with distinct roots into the canonic form $z^3 - 1 = 0$ is elegant and intuitive, but not the only way to solve the cubic. For example, we could Möbius transform the general cubic with distinct roots into a palindromic cubic of the form $Az^3+Bz^2+Bz+A = (z+1)(Az^2+(B-A)z+A)$, and thereby reduce the equation to a quadratic. Alternatively, we could Möbius transform the general cubic into the equation $(z+1)z(z-1) = z^3-z$. This method is especially interesting when the roots of the cubic are known *a priori* to be real,[12] because in this case the Möbius coefficients are all real.

Solving a quartic (biquadratic) polynomial

Quartic polynomials have 4 roots. Since $z^4 - 1$ has all 4 roots $(\pm 1, \pm i)$ on the unit circle, and since one can construct quartic polynomials whose roots are not on a circle, we cannot hope to use just a Möbius transformation to transform an arbitrary quartic polynomial into the canonic form $z^4 - 1$. However, we may be able to utilize enough of our insight gained from the cubic case to still solve the quartic.

We recall that in the cubic case, we utilized a Möbius transformation to produce a cubic polynomial in which the two internal coefficients (a_2 and a_1) were both zero.[13] We avoided a substantial amount of algebraic manipulation by using Taylor series to extract the expressions for these internal coefficients from the Möbius-transformed polynomial.

We will utilize the same Möbius transform technique on the quartic polynomial $p(z) = z^4 + 0z^3 + a_2z^2 + a_1z + a_0$ (notice that we have already translated to the center-of-mass of the 4 roots to assure $a_3 = 0$) to produce a polynomial $q(y = b_4y^4 + 0y^3 + b_2y^2 + 0y + b_0$ — i.e., the *odd* coefficients b_3 of y^3 and b_1 of y are both *zero*. The polynomial $q(y)$ produced by this Möbius transformation is now just a quadratic equation in disguise — $q(y) = r(y^2)$. We can then solve this quadratic for 2 values, the square roots of which become 4 values. These 4 values are then Möbius-transformed into the roots of the original quartic equation.

In more detail, we construct the equations to guarantee that $b_3 = 0$ and $b_1 = 0$:

$$4p(r) - (r-s)p'(r) = 0$$
$$4p(s) + (r-s)p'(s) = 0$$

As before, if we subtract the first from the second, the resulting equation is obviously divisible by $r - s$:

$$2rs(r+s) + a_2(r+s) + a_1 = 0$$
$$2a_2rs(r+s) + a_1(r+s)^2 - 6a_1rs + 4a_0(r+s) = 0$$

Solve the first equation for rs, and then substitute into the second. After clearing fractions, we get a *cubic* equation in the variable $r + s$. Solve this equation using the techniques above, and then substitute back into the first equation to get rs. Finally, construct the *quadratic* equation $x^2 - (r+s)x + rs = 0$, and solve it for r, s, as in the case of the cubic. We now utilize this Möbius transform to produce the quartic/quadratic $r(y^2)$, which quadratic is again solved using the techniques above. Thus, our method for the quartic requires the solution of a cubic plus 2 quadratics, or if we also count the quadratic used in the solution of the cubic, we require in total a cube root extraction plus 3 square root extractions to solve the quartic.

As in the case of the cubic, there may be other ways to utilize Möbius transforms to solve the quartic. For example, it might be possible to Möbius-transform a general quartic into a palindromic quartic, in which case palindromic techniques would allow a reduction to a quadratic.

Conclusions

We found that we could use powerful geometric transformation techniques to produce canonic forms for polynomials of degree 1, 2, and 3. For degree 4, we could not produce a canonic form, but were nontheless able to reduce the quartic into a quadratic using a Möbius transformation analogous to that for transforming the cubic into canonic form. Furthermore, given a general polynomial $p(z)$ of degree n, this particular Möbius transformation can be used to produce a new polynomial $q(z)$ of degree n, but the coefficients of z^{n-1} and z in $q(z)$ are both zero. Furthermore, the parameters of this Möbius transformation can be found by solving a polynomial of degree $n-1$. Thus, it should be possible to transform any *quintic* polynomial into the form $z^5 + Pz^3 + Qz^2 + R$ by solving a *quartic* equation. However, our Möbius techniques cannot be applied to further reduce this quintic polynomial.[14]

[12] For example, the cubic equation could be the characteristic equation of a 3x3 *Hermitian* matrix.

[13] We thus produced a "bicoastal" polynomial.

[14] Since it is easy to transform the coefficients of z^n, z^0 to one, and the coefficients of z^{n-1} and z^1 to zero, the coefficients in these "outer

References

[Adobe90] Adobe Systems, Inc. *Postscript Language Reference Manual, 2nd Ed*. Addison-Wesley, Reading, MA, 1990.

[Altmann86] Altmann, S.L. *Rotations, Quaternions, and Double Groups*. Clarendon Press, Oxford, 1986.

[Autodesk92] Autodesk, Inc. *AutoCAD Release 12 Customization Manual*. Autodesk, Inc., Sausalito, CA, 1992.

[Ball08] Ball, W.W.R. *A Short Account of the History of Mathematics*. Dover Publ., 1908.

[Beeler72] Beeler, M, Gosper, R.W., and Schroeppel, R. *HAKMEM*. MIT AI Memo No. 239, Feb. 29, 1972.

[Barbeau89] Barbeau, E.J. *Polynomials*. Springer-Verlag, New York, 1989.

[Brown88] Brown, James A., *et al*. *APL2 at a Glance*. Prentice Hall, Englewood Cliffs, NJ, 1988.

[Cardano68] Cardano, G. *Artis Magnae, sive de regulis algebraicis* ("The Great Art, or The Rules of Algebra"). 1545. Transl. Witmer, T.R., MIT Press, Cambridge, MA 1968.

[Cayley76] Cayley, A. "Synopsis of the Theory of Equations". *Messenger of Math. v* (1876), 39-49.

[Cayley79q] Cayley, A. "Application of the Newton-Fourier Method to an Imaginary Root of an Equation". *Quart. J. Pure & Appl. Math.*, xvi (1879), 179-185.

[Cayley79c] Cayley, A. "The Newton-Fourier imaginary problem". *Amer. J. Math. II*, 97 (1879).

[Curry83] Curry, J., Garnett, L., and Sullivan, D. "On the iteration of rational funtions: Computer experiments with Newton's method". *Commun. Math. Phys. 91* (1983), 267-277.

[Dejon67] Dejon, B., and Henrici, P., *Eds*. *Constructive Aspects of the Fundamental Theorem of Algebra*. Wiley, London, 1967.

[EB94] EB. *Encyclopaedia Britannica, 15th Edition*. 1994. *Britannica Online*: `http://www.eb.com/`

shells" are "less tightly bound" than the inner coefficients, to make an analogy with electron shells in the atomic chemistry of the elements.

[Friedman86] Friedman, J. "On Newton's Method for Polynomials". *IEEE 27th Symp. on FOCS*, Oct. 1986, 153-161.

[Gargantini72] Gargantini, I., and Henrici, P. "Circular Arithmetic and the Determination of Polynomial Zeros". *Numer. Math. 18* (1972), 305-320.

[Garver27] Garver, R. "Tschirnhaus Transformations on Certain Rational Cubics". *Amer. Math. Monthly XXXIV*, 10 (Dec. 1927), 521-525.

[Garver29a] Garver, R. "Linear Fractional Transformations on Quartic Equations". *Amer. Math. Monthly XXXVI*, 4 (Apr. 1929), 208-212.

[Garver29b] Garver, R. "Transformations on Cubic Equations". *Amer. Math. Monthly XXXVI*, 7 (Aug.-Sep. 1929), 366-369.

[Garver31] Garver, R. "Two Applications of Tschirnhaus Transformations in the Elementary Theory of Equations". *Amer. Math. Monthly XXXVIII*, 4 (Apr. 1931), 185-188.

[Graber09] Graber, M.E. "Note on the General Quartic". Amer. Math. Monthly XVI (1909), 27-29.

[Henderson30] Henderson, A., and Hobbs, A.W. "The Cubic and Biquadratic Equations: Vieta's Transformation in the Complex Plane". *Amer. Math. Monthly XXXVII*, 10 (Dec. 1930), 515-521.

[Knuth81] Knuth, D.E. *Seminumerical Algorithms, 2nd Ed*. Addison-Wesley, Reading, MA, 1981.

[Klein56] Klein, Felix. *Lectures on the Icosahedron and the Solution of Equations of the Fifth Degree*. Morrice, G.G., transl. from 1888, 1913 manuscript. Dover Publs. Inc., New York, 1956.

[Lovitt39] Lovitt, W.V. *Elementary Theory of Equations*. Prentice-Hall, New York, 1939.

[Mandelbrot83] Mandelbrot, B.B. *The Fractal Geometry of Nature, 2nd Ed*. W.H. Freeman and Co., New York, 1983.

[Peitgen86] Peitgen, H.-O., and Richter, P.H. *The Beauty of Fractals: Images of Complex Dynamic Systems*. Springer-Verlag, Berlin, 1986.

[Schwerdtfeger79] Schwerdtfeger, H. *Geometry of Complex Numbers*. Dover Publ., New York, 1979.

[Turnbull46] Turnbull, H.W. *Theory of Equations*. Oliver and Boyd, Edinburgh and London; New York: Interscience Publishers, Inc., 1946.

[Zippel93] Zippel, R. *Effective Polynomial Computation*. Kluwer Academic Publishers, Boston, 1993.

Henry Baker has been diddling bits for 35 years, with time off for good behavior at MIT and Symbolics. In his spare time, he collects garbage and tilts at windbags. *This column appeared in ACM Sigplan Notices 33,3 (Mar 1998), 24-35.*